Journal of Peridynamics and Nonlocal Modeling (2022) 4:159–200
https://doi.org/10.1007/s42102-021-00072-z

ORIGINAL ARTICLES

Peridynamics for Data Estimation, Image Compression/ Recovery, and Model Reduction

Erdogan Madenci[1] · Atila Barut[1] · Evan Willmarth[2] · Nam Phan[3]

Received: 1 April 2021 / Accepted: 8 November 2021 / Published online: 4 January 2022

Abstract
The existing interpolation and regression methods are highly data-specific, challenge-specific, or approach-specific. Peridynamic approach provides a single mathematical framework for diverse data-sets and multi-dimensional data manipulation and model order reduction. The mathematical framework based on the Peridynamic Differential Operator (PDDO) provides a unified approach to transfer information within a set of discrete data, and among data sets in multi-dimensional space. The robustness and capability of this approach have been demonstrated by considering various real or fabricated data concerning two- or three-dimensional applications. The numerical results concern interpolation of real data in two and three dimensions, interpolation to approximate a three-dimensional function, adaptive data recovery in three-dimensional space, recovery of missing pixels in an image, adaptive image compression and recovery, and free energy evaluation through model reduction.

Keywords Peridynamic · Interpolation · Regression · Data · Image · Compression · Recovery · Model reduction

1 Introduction

The domain-agnostic mathematical representations and datafication require interpolation and regression of discrete data. Interpolation and regression of data play a significant role in many scientific disciplines. A data point, entity, object, etc. interacts with its

✉ Erdogan Madenci
madenci@email.arizona.edu

Atila Barut
abarut555@gmail.com

Evan Willmarth
evan.willmarth@yale.edu

Nam Phan
nam.phan@mil.gov

[1] The University of Arizona, Tucson, AZ 85721, USA

[2] Yale University, New Haven, CT 06520, USA

[3] US Naval Air Systems Command, Patuxent River, MD 20670, USA

surrounding media. In real life, data sets are mostly connected with surge of irregularities, breakages, and discontinuities. It is essential that interpolation and regression methods capture such irregularities and scatter within a set of discrete data in a multi-dimensional space. Also, they should transfer information among data sets representing multiple scales such as fine and coarse models.

Interpolation is an estimation of an unknown variable at output points (locations) by employing the known values at surrounding input points. Regression is an estimation of a variable at both input and output points by employing the known values at the surrounding input locations. Smoothing is an estimation of a variable at only known input points by employing the known input values. Smoothing may be necessary if the input data is noisy. The estimation of the unknown variable is based on interpolation passing through all the known input values. In other words, there is an exact recovery of the known values of the input points.

Although the idea of interpolation and regression seems to be rather rudimentary, it has a profound impact and forms one of the building blocks in science and engineering. Over the years, scientists have tried to elaborate on it through different methods as discussed by Franke [1], Mittas and Mitasova [2], and Steffensen [3].

The simplest interpolation method is the polynomial expansion. It requires the determination of the coefficients of a complete polynomial by using the known input values. The coefficients are determined such that the polynomial recovers the known values at the input points. Hence, a system of equations is generated to solve for the unknown coefficients. Although the polynomial form of interpolation is simple to apply, it is not practical if the number of input values is substantially high.

The Lagrangian functions can be employed to eliminate the process of solving a large system of equations. This approach generates a unique set of polynomials for each input point such that it is equal to the input value at the input point, and zero at all other points. Combination of the Lagrangian functions forms the interpolation. However, the input points must form a structured grid especially in two-dimensional applications in order generate a unique set of Lagrangian polynomials. As the number of input points increases, both the polynomial and Lagrangian forms of interpolations require high degree of polynomials leading to undesirable oscillations.

An alternative to the polynomial and Lagrangian interpolations is the spline interpolation in which a low-order polynomial passes through the adjacent points. The spline interpolation ensures the continuity of the polynomials at the input points. However, it also requires a structured grid configuration in two-dimensional applications; thus, it cannot be applied to scattered data points. Additional conditions may have to be imposed based on the nature of interpolation technique and the character of the data describing the phenomenon. As discussed by Cressie [4], these conditions may be based on geostatistical concepts (Kriging), locality (nearest neighbor), smoothness (splines), or functional forms (polynomials). These techniques yield satisfactory predictions for smooth variations without scatter.

Liszka and Orkisz [5] and Liszka [6] introduced a method to consider scattered data without using high degree of polynomials. The interpolation is achieved by employing the Taylor Series Expansion (TSE) about the output points. The TSE is truncated after the second order derivative terms. For each output point, a system of equations is established by enforcing the function (i.e., TSE) to match the value at the input points. Thus, it leads to a set of equations to solve for the unknowns at the output points. However, the values of the function at the output points are obtained via least squares minimization because the number of equations in the resulting system is more than the unknowns at the output points.

This study introduces a unified approach for interpolation and regression of data with irregularities and scatter in a multi-dimensional space based on the non-local Peridynamic Differential Operator (PDDO) within a set of discrete data and among data sets representing multiple scales. Also, the PD interpolation functions between fine- and coarse-level grids enable the reduction of number of unknowns in the analysis while retaining the accuracy associated with the fine grid.

The most common applications of interpolation and regression analyses with the existing methods are conducted in one- and two-dimensional spaces. The PDDO is not limited by the order of dimension. Also, the present approach is not restricted to any kind of spatial discretization. In addition, it is not limited for interpolating fields with C^0 continuity. Furthermore, the evaluation of the determinant of very large matrices is not tractable due to memory, precision, and computational time requirements. The PD interpolation functions between the unknowns of the fine and coarse grids enable the reduction of the size of stiffness matrix and yet retain sufficient accuracy. It is worth noting that the PDDO is computationally more expensive than the existing methods when using a single processor; however, it is extremely suitable for GPU architecture. The computational speed will be the topic of a future study.

Subsequent sections present the PDDO operator and its use for interpolation and regression within a scale and between fine and coarse scales. The numerical results concern interpolation of real data in two and three dimensions, interpolation to approximate a three-dimensional function, adaptive data recovery in three-dimensional space, recovery of missing pixels in an image, adaptive image compression and recovery, and accurate evaluation of free energy. The model reduction is achieved by employing PD interpolation between fine and coarse grids. It is demonstrated by considering the thermal fluctuation of a rod.

2 Peridynamic Differential Operator

Recently, Madenci et al. [7] introduced the Peridynamic Differential Operator (PDDO) to approximate the non-local representation of a scalar field $f = f(\mathbf{x})$ and its de'rivatives at point $\mathbf{x}$ by accounting for the effect of its interactions with the other points, $\mathbf{x'}$, in the domain of interaction, as shown in Fig. 1.

The PDDO employs the concept of PD interactions, and the PD functions without performing any differentiation as explained by Madenci et al. [7, 8]. The PDDO requires the construction of PD functions. They are determined directly by making them orthogonal to each term in the Taylor Series Expansion (TSE). The derivation of the PDDO up to second-order derivatives of a function with two independent variables is presented in Appendix 1.

The major difference between the PDDO and other existing local and non-local numerical differentiation methods is that the PDDO leads to analytical expressions for arbitrary order derivatives in integral form for symmetric interaction domains. It provides accurate estimation of the function and its derivatives in the interior as well as the near the boundaries of the domain without any special boundary treatment. Also, the PD differentiation serves as a natural filter in the presence of noisy data, as it provides their derivatives [7, 8].

Each point has its own family members in the domain of interaction (family), and occupies an infinitesimally small entity such as volume, area, or a distance. The points $\mathbf{x}$ and $\mathbf{x'}$ only interact with the other points in their own families, H_x and $H_{x'}$, respectively.

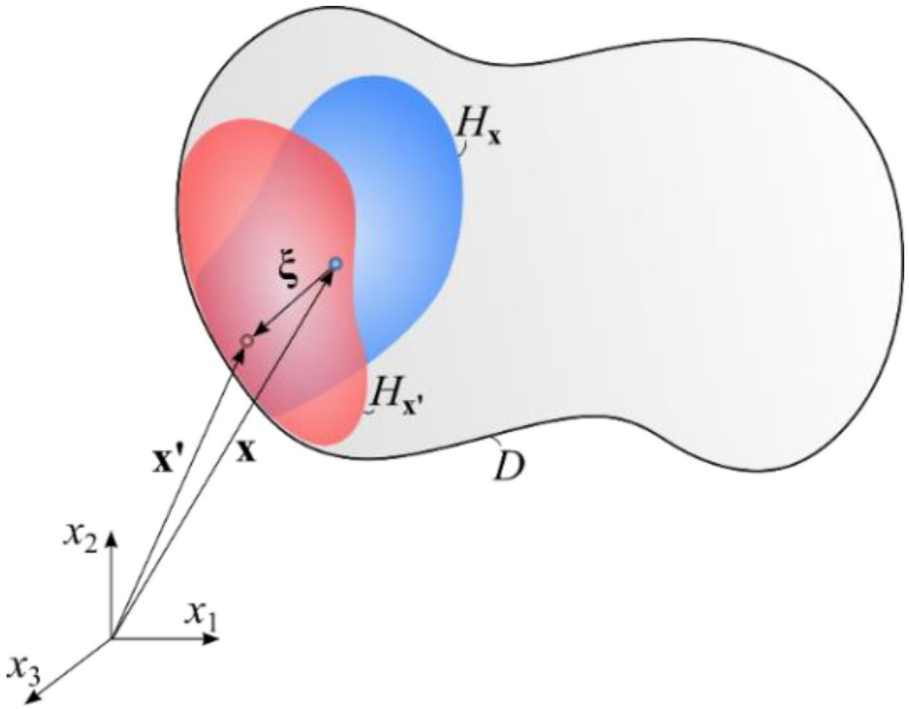


Fig. 1 Interaction of peridynamic points, $\mathbf{x}$ and $\mathbf{x}'$, with arbitrary family size and shape

Neither point $\mathbf{x}$ nor $\mathbf{x}'$ is necessarily symmetrically located in their interaction domains. The initial relative position, ξ, between the material points $\mathbf{x}$ and $\mathbf{x}'$ can be expressed as $\xi = \mathbf{x}' - \mathbf{x}$. This ability permits each point to have its own unique family with an arbitrary position. Therefore, the size and shape of each family can be different, and they significantly influence the degree of non-locality. The degree of interaction between the material points in each family is specified by a non-dimensional weight function, $w(|\xi|)$ which can vary from point to point. The interactions become more local with decreasing family size. Thus, the family size and shape are important parameters. In general, the family of a point can be non-symmetric due to non-uniform spatial discretization. The PDDO is not restricted to any kind of spatial discretization. The family points can be uniformly or arbitrarily spaced. Thus, the number of family members can vary depending on the discretization. The family members of point $\mathbf{x}$ can be selected by simply retaining the neighboring points within a circle or by applying a family search method such as KD tree and clustering.

The PDDO for the N-th order derivative of a function $f(\mathbf{x})$ with M dimensions can be expressed as

$$\frac{\partial^{p_1+p_2+\cdots+p_N} f(\mathbf{x})}{\partial x_1^{p_1} \partial x_2^{p_2} \cdots \partial x_M^{p_N}} = \int_{H_{\mathbf{x}}} f(\mathbf{x}+\boldsymbol{\xi}) g_N^{p_1 p_2 \cdots p_N}(\boldsymbol{\xi}) d\xi_1 d\xi_2 \cdots d\xi_M \tag{1}$$

in which p_i denotes the order of differentiation with respect to variable x_i with $i = 1, \ldots, M$, and $g_N^{p_1 p_2 \cdots p_N}(\boldsymbol{\xi})$ are the PD functions explained in detail in a recent book by Madenci et al. [8].

They can be constructed as

$$g_N^{p_1 p_2 \cdots p_N}(\boldsymbol{\xi}) = \sum_{q_1=0}^{N} \sum_{q_2=0}^{N-q_1} \cdots \sum_{q_N=0}^{N-q_1 \cdots - q_{N-1}} a_{q_1 q_2 \cdots q_N}^{p_1 p_2 \cdots p_N} \omega_{q_1 q_2 \cdots q_N}(|\boldsymbol{\xi}|) \xi_1^{q_1} \xi_2^{q_2} \cdots \xi_M^{q_N} \tag{2}$$

where $\omega_{q_1 q_2 \cdots q_N}(|\boldsymbol{\xi}|)$ is the weight function associated with each term $\xi_1^{q_1} \xi_2^{q_2} \cdots \xi_M^{q_N}$ in the polynomial expansion. The PDDO recovers the local differentiation as the size of family $H_{\mathbf{x}}$ decreases or the number of terms in the functions $g_N^{p_1 p_2 \cdots p_N}(\boldsymbol{\xi})$ increases.

The coefficients of the PD functions can be determined without any difficulty. Although it is not a limitation, the weight functions, $\omega_{q_1 q_2 \cdots q_N}(|\boldsymbol{\xi}|)$, in Eq. (2) can be replaced with $\omega_n(|\boldsymbol{\xi}|)$ for simplification based on the order of differentiation. A MATLAB code for performing PD differentiation for the N-th order derivative of a function with M dimensions is given by Madenci et al. [8].

3 Peridynamics for Estimation

This section provides the construction of functions for PD interpolation and regression. It is a unique framework for data estimation and data recovery. The PD interpolation estimates the unavailable data from the available data set while passing through all the available data. However, the PD regression does not necessarily pass through all the input points. The input data referred to as available data points can be uniformly or arbitrarily spaced without any restriction to spatial discretization. Therefore, the number of family members for each point can be different depending on the nature of the data. The contribution of each available data point is distributed to the unknown data points based on an area based fraction parameter.

As shown in Fig. 2, there may exist M input points with respect to a Cartesian coordinate system. Each input point $\tilde{\mathbf{x}}_j = \mathbf{x}_k + \boldsymbol{\xi}_{kj}$ occupies a volume of $\tilde{V}_j$, and a generic (output) point $\mathbf{x}_k$ occupies a volume of V_k. The PD interaction domain (family) of the generic point $\mathbf{x}_k$ is H_k. While the shape of its interaction domain can be arbitrary, the output point, $\mathbf{x}_k$, has a horizon size of δ_k which represents the radius of a sphere encompassing a specified number of input points as shown in Fig. 2. It defines the family population, N_k, of the output point, $\mathbf{x}_k$, i.e., number of input points, $\tilde{\mathbf{x}}_j$, in H_k. The specified weight function, $\omega_{kj} = \omega(\left|\boldsymbol{\xi}_{kj}\right|)$, dictates the influence of the input points on the output points.

The value of the function at the input point, $(\tilde{x}_j, \tilde{y}_j)$, is denoted by $\tilde{f}_j = \tilde{f}(\tilde{\mathbf{x}}_j)$, for $j = 1, \ldots, M$. As derived in Appendix 1, the PD approximation of the function (zeroth-order derivative) at point, $\mathbf{x}_k$, can be expressed in discrete form as

$$f(\mathbf{x}_k) \cong \sum_{j=1}^{N_k} \tilde{f}(\mathbf{x}_k + \boldsymbol{\xi}_{kj}) g_2^{000}\left(\boldsymbol{\xi}_{kj}; \omega_{kj}, \tilde{V}_j\right) \tilde{V}_j \tag{3}$$

In matrix form, it can be rewritten as

$$f\left(\mathbf{x}_k\right) = \mathbf{h}^T\left(\boldsymbol{\xi}_{kj}\right) \tilde{\mathbf{f}}\left(\tilde{\mathbf{x}}_j\right) \text{with } j = 1, \ldots, M \tag{4}$$

where $\mathbf{h}$ is the vector of PD estimation and $\tilde{\mathbf{f}}$ is the vector of input values associated with point $\mathbf{x}_k$. They are defined as

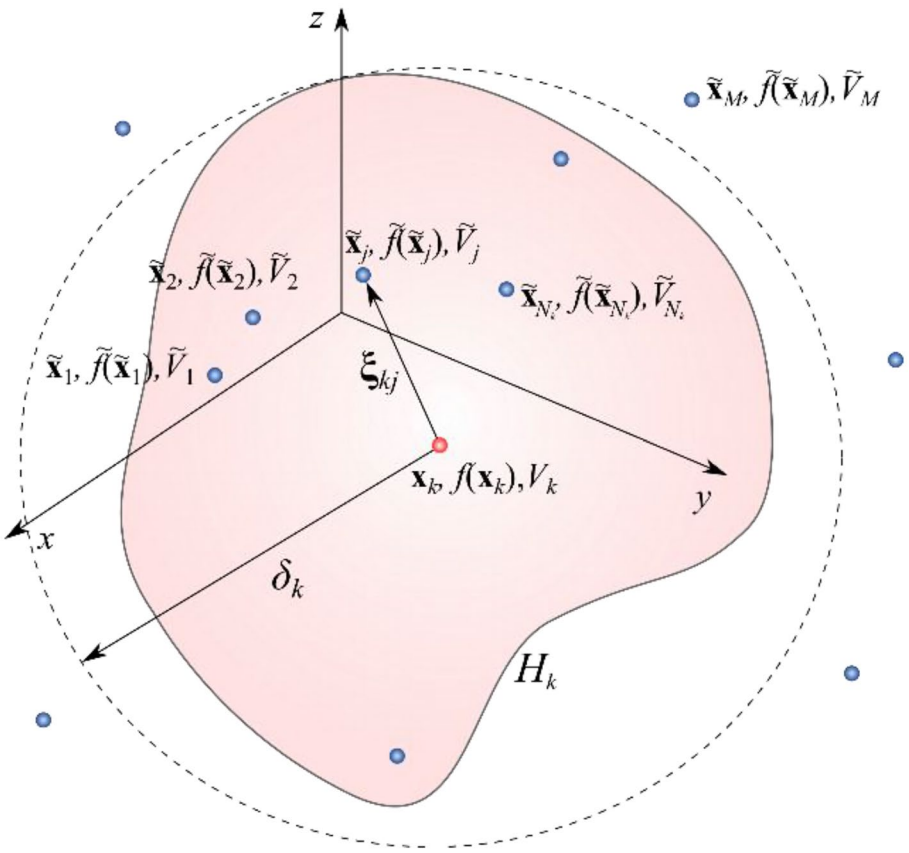


Fig. 2 Input points within the family of output point, $\mathbf{x}_k$

$$\mathbf{h}^T\left(\boldsymbol{\xi}_{kj}\right) = \left\{g_2^{000}\left(\boldsymbol{\xi}_{kj};\omega_{k1},\tilde{V}_1\right)\tilde{V}_1,\ldots,g_2^{000}(\boldsymbol{\xi}_{kN_k};\omega_{kN_k},\tilde{V}_{N_k})\tilde{V}_{N_k}\right\} \tag{5}$$

and

$$\tilde{\mathbf{f}}^T(\tilde{x}_j) = \left\{\tilde{f}_1,\tilde{f}_2,\ldots,\tilde{f}_{N_k}\right\}. \tag{6}$$

where $\xi_{kj} = \tilde{\mathbf{x}}_j - \mathbf{x}_k$ and the subscript 2 represent the highest order of derivatives retained in the TSE. The derivation of the PD function, $g_2^{000}\left(\xi_{kj};\omega_{kj},\tilde{V}_j\right)$, is described in Appendix 1.

The PD approximation given by Eq. (3) passes through all input points within the horizon of point $\mathbf{x}_k$ for

$$\omega_{kj} = \left(\frac{\delta_k}{\xi_{kj}}\right)^p \text{ with } p > 1. \tag{7}$$

Thus, it is referred to as the PD interpolation, and it can be applied to estimate the missing (unavailable) data from the existing (available) data set.

The PD approximation given by Eq. (3) does not necessarily pass through all the input points for any other form of a weight function such as

$$\omega_{kj} = e^{-4\xi_{kj}^2/\delta_k^2} \tag{8}$$

This PD approximation provides a regression (curve fit) through the input points. As demonstrated by Madenci et al. [7, 8], it can be also used to filter the noise and smooth out the irregularities in the data. The major difference between these weights is that $\omega_{kj} = e^{-4\xi_{kj}^2/\delta_k^2}$ approaches a unit value whereas $\omega_{kj} = \delta_k^p/\xi_{kj}^p$ approaches infinity as $|\xi_{kj}| \to 0$. As the number of spacing and the horizon size decreases, the degree of interaction becomes stronger, and the PD regression recovers the interpolation.

It is worth pointing out that Eq. (3) is not limited to a three-dimensional space; it is expandable to higher dimensional spaces as derived by Madenci et al. [8]. However, the literature shows that most common applications of interpolation and regression analyses are conducted in one- and two-dimensional spaces.

As shown in Fig. 3, the output point (x_k, y_k) associated with area, A_k, and the corresponding recovered data $f(x_k, y_k)$ for $k = 1, .., K$ are denoted by blue circles.

A set of M input points are arbitrarily positioned on the $x - y$ plane. The two-dimensional form of the vector of PD interpolation or regression function becomes

$$\mathbf{h}^T(\xi_{kj}) = \left\{ g_2^{00}(\xi_{k1}; w_{k1}, \tilde{A}_1)\tilde{A}_1, .., g_2^{00}(\xi_{kN_k}; w_{kN_k}, \tilde{A}_{N_k})\tilde{A}_{N_k} \right\} \tag{9}$$

in which ξ_{kj} denotes the relative position vector between input and output points and $\tilde{A}_j$ represents the area associated with each input point, $(\tilde{x}_j, \tilde{y}_j)$, which are yet unknown. The relative position vector, ξ_{kj}, is defined as

$$\xi_{kj} = \left\{ (\tilde{x}_j - x_k), (\tilde{y}_j - y_k) \right\}^T \tag{10}$$

with

$$\xi_{kj} = \sqrt{\left(\tilde{x}_j - x_k\right)^2 + \left(\tilde{y}_j - y_k\right)^2} \tag{11}$$

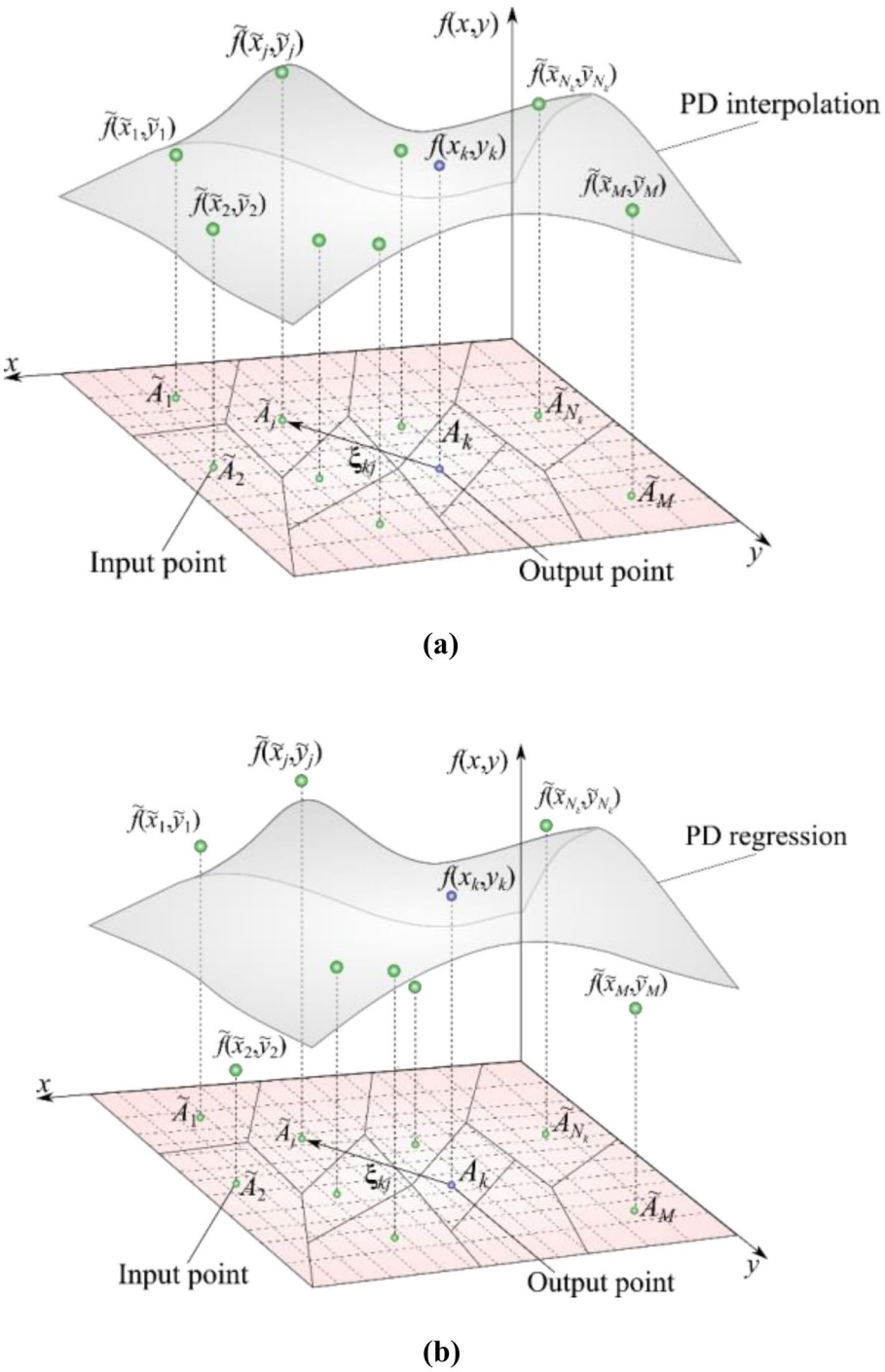
As shown in Fig. 4, the area of each output point, A_k, is rectangular in shape. The output points are located at the center of each area defined by $A_k = (x_k - x_{k-1})(y_k - y_{k-1})$, and both the locations of output points and their associated areas, A_k, are known. It must be kept in mind that the total area of input points must be identical to that of output points. Hence, the total area, A, is defined in terms of the sum of the areas associated with the output points as

$$A = \sum_{k=1}^{K} A_k \tag{12}$$

Similarly, the total area of the domain can be computed from the sum of the areas of input points as

$$A = \sum_{j=1}^{M} \tilde{A}_j \tag{13}$$

There is no unique way to express the unknown areas of input points in terms of those of output points. However, the unknown area of each input point $\tilde{A}_j$ can be estimated by the weighted distribution of the area of an output point, A_k, to all of the areas



(a)

(b)

Fig. 3 Input and output point for two-dimensional: (**a**) interpolation and (**b**) regression

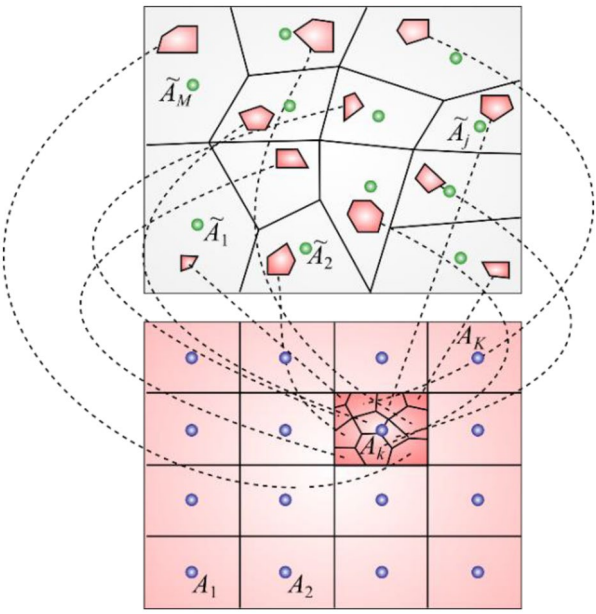


Fig. 4 Distribution of area segment from an output point to the input points

of input points as shown in Fig. 4. To achieve a weighted distribution, a fraction parameter ρ_{kj} is defined as

$$\rho_{kj} = \frac{A_{kj}}{A_k} \tag{14}$$

in which A_{kj} represents the segment of A_k distributed to $\tilde{A}_j$. Note that the fraction parameter ρ_{kj} varies between $0 \leq \rho_{kj} \leq 1$ and it satisfies the partition of unity, i.e., $\sum_{r=1}^{M} \rho_{kr} = 1$. In accordance with this assumption, A_k can be expressed as

$$A_k = \sum_{j=1}^{M} A_{kj} \tag{15}$$

Substituting from Eqs. (15) and (12) into Eq. (13) leads to

$$\sum_{j=1}^{M} \tilde{A}_j = \sum_{k=1}^{K} \sum_{j=1}^{M} A_{kj} \tag{16}$$

or

$$\sum_{j=1}^{M} \left(\tilde{A}_j - \sum_{k=1}^{K} A_{kj} \right) = 0 \tag{17}$$

After substituting from Eq. (14), this equation yields the expression for $\tilde{A}_j$ in the form

$$\tilde{A}_j = \sum_{k=1}^{K} \rho_{kj} A_k \tag{18}$$

The fraction parameter, ρ_{kj}, can be defined as ratio of the weight defined between an input and output point, $\omega_{kj} = \delta_k^p / \xi_{kj}^p$, to the sum of the weights defined between the same input and all output points as

$$\rho_{kj} = \frac{\omega_{kj}}{\sum_{r=1}^{M} \omega_{kr}} = \frac{1/\xi_{kj}^p}{\sum_{r=1}^{M} 1/\xi_{kr}^p} \tag{19}$$

Invoking Eq. (19) into Eq. (18) provides the value of $\tilde{A}_j$ in terms of A_k as

$$\tilde{A}_j = \sum_{k=1}^{K} \left(\frac{1/\xi_{kj}^p}{\sum_{r=1}^{M} 1/\xi_{kr}^p} \right) A_k \text{ for } j = 1, M \tag{20}$$

in which $p \geq 1$. This approximation satisfies the requirement of conservation of area. For a uniform spacing among the output points, the area of each output point, A_k, can be set to $A_k = A/K$. Hence, Eq. (20) simplifies to

$$\tilde{A}_j = \frac{A}{K} \sum_{k=1}^{K} \left(\frac{1/\xi_{kj}^p}{\sum_{r=1}^{M} 1/\xi_{kr}^p + \varepsilon} \right) \text{for } j = 1, M \tag{21}$$

in which K denotes the number of output points in the region. Also, a small number, $\varepsilon = 10^{-9}$, is introduced in order eliminate any possible numerical singularity.

4 Peridynamic Regression for Data Compression and Recovery

The PD regression can be applied to data compression and recovery by employing a selective set of data with known values as input points, referred to as picked data, from the original data set. By employing Eq. (4), the unknown data points can be estimated based on the known (or picked) data by

$$f(\mathbf{x}_k) = \mathbf{h}^T(\xi_{kj}) \tilde{\mathbf{f}}(\tilde{\mathbf{x}}_j) \tag{22}$$

This equation enables the recovery of data points by using only a portion of the data with N_p points from the original data set with N points. It can be rewritten as

$$f(\mathbf{x}_k) = \sum_{j=1}^{N_p} H(\delta - \xi_{kj}) \omega_{kj} g_2^{00}(\xi_{kj}) \tilde{f}(\tilde{\mathbf{x}}_j) \tilde{A}_j \quad (k = 1, \ldots, N_u) \tag{23}$$

where N_p and N_u indicate the number of picked and unpicked points, $H(\delta - \xi_{kj})$ is the Heaviside step function, and weight function ω_{kj} is defined as

$$\omega_{kj} = e^{-4\xi_{kj}^2/\delta_k^2} \tag{24}$$

Note that the total of the number of picked and unpicked points are equal to the total number of points in the original data set, i.e., $N = N_p + N_u$.

The complete data set with N sample points can be stored in a vector, $\mathbf{f}^*$, as

$$\mathbf{f}^* = \left\{f_1^*, f_2^*, \ldots, f_N^*\right\}^T \tag{25}$$

The iterative for procedure adaptive data compression and recovery involves the following steps:

1. Start by randomly picking 1% of the N data points of the original data and store them in the vector, $\tilde{\mathbf{f}}$, as

$$\tilde{\mathbf{f}}^T = \{f_{p_1}^*, f_{p_2}^*, \ldots, f_{p_{N_p}}^*\} \tag{26}$$

where p_k with $(k = 1, \ldots, N_p)$ represents the index IDs of the picked data and N_p is the number of picked data points. The remaining unpicked data points are unknown and they are contained in vector, $\dot{\mathbf{f}}$, as

$$\dot{\mathbf{f}}^T = \{f_{u_1}, f_{u_2}), \ldots, f_{u_{N_u}}\} \tag{27}$$

where u_k with $(k = 1, \ldots, N_u)$ denotes the index ID numbers of the unknown data points with N_u representing the number of unknown data points.
2. Use Eq. (23) along with Eq. (21) to predict the unknown data in vector, $\dot{\mathbf{f}}$.
3. Compute the difference between the original and the predicted data values of the unpicked data points as

$$\Delta f_{u_k} = \left|f_{u_k}^* - f_{u_k}\right| (k = 1, \ldots, N_u) \tag{28}$$

4. Sort the difference values, Δf_{u_k}, in descending order and store them in a new array, $\Delta\dot{\mathbf{f}}$, defined as

$$\Delta\dot{\mathbf{f}}^T = \left\{\Delta f_{n_1}, \Delta f_{n_2}, \ldots, \Delta f_{n_{N_u}}\right\} \tag{29}$$

where the indices n_k with $(k = 1, \ldots, N_u)$ are ordered such that

$$\Delta f_{n_1} > \Delta f_{n_2} > \cdots > \Delta f_{n_{N_u}} \tag{30}$$

5. Calculate the error, e, due to Δf_{n_k} defined in the form

$$e = \frac{1}{\left|f_{\max}^*\right|}\sqrt{\frac{1}{N}\sum_{j=1}^{N_u}\left(\Delta f_{n_j}\right)^2} \tag{31}$$

where $\left|f_{\max}^*\right|$ is the maximum absolute value of $\mathbf{f}^*$ among N data points and it is defined as

$$|f^*_{\max}| = \text{Max}\left\{ |f^*_1|, |f^*_2|, \dots, \left|f^*_N\right| \right\} \tag{32}$$

6. If $e < 0.1$, convergence is achieved; print the converged results for unpicked data points along with the known data points and stop the analysis. Otherwise, continue with step 7.
7. Pick additional M points from the original data set, $\mathbf{f}^*$, and append them to the vector of picked data set, $\tilde{f}$, as

$$\tilde{\mathbf{f}}^T = \left\{ f^*_{p_1}, f^*_{p_2}, \dots, f^*_{p_{N_p}}, f^*_{n_1}, f^*_{n_2}, \dots, f^*_{n_M} \right\} \tag{33}$$

   Limit the size of M not more than 5% of the total number of data points, N. The indices for these additional data points are chosen from the first M indices of vector $\Delta\mathbf{f}$ in Eq. (29) for faster convergence. Subsequently, remove the data points with indices n_1 through n_M from the vector of unpicked data points, in Eq. (27), to balance the total of picked and unpicked data points, in the form

$$\mathbf{f}^T = \left\{ f_{(n_{M+1})}, f_{(n_{M+2})}, \dots, f_{(n_{N_u})} \right\} \tag{34}$$

   Also, update the number of picked and unpicked data points as $N_p = N_p + M$ and $N_u = N_u - M$.
8. Continue performing steps 2 through 8 until convergence is reached in step 6.

5 Peridynamic Regression for Image Compression and Recovery

The PD regression can be applied to image compression and recovery by employing a selective set of pixels with known values as input points, referred to as picked pixels, from the original image.

An image is described by a set of pixels each of which includes three basic color tones, known as the RGB which stands for Red, Green, and Blue. These colors usually vary between 0 and 255. Combination of varying color tones of red, green, and blue provide the true color of the pixel. For example, pure white color is achieved by RGB $= (255, 255, 255)$ and pure black color by RGB $= (0, 0, 0)$. As shown in Fig. 5, H and W denote the height and width of the image, respectively. The coordinates of a pixel with an unknown value, P_k, on the image are defined by x_k and y_k. Its unknown RGB values are defined by $r_k(x_k, y_k), g_k(x_k, y_k)$ and $b_k(x_k, y_k)$, respectively. Similarly, the coordinates of a pixel with a known value, $\tilde{P}_j$, on the image are defined by $\tilde{x}_j$ and $\tilde{y}_j$. Its known (available) RGB pixel values are defined by $\tilde{r}_j(\tilde{x}_j, \tilde{y}_j), \tilde{g}_j(\tilde{x}_j, \tilde{y}_j)$ and $\tilde{b}_j(\tilde{x}_j, \tilde{y}_j)$.

By employing Eq. (4), the unknown RGB values at pixel P_k can be estimated based on the known (or picked) pixels by

$$f_c(\mathbf{x}_k) = \mathbf{h}^T_c(\xi_{kj})\tilde{\mathbf{f}}_c(\tilde{\mathbf{x}}_j) \tag{35}$$

in which the subscripts $c = r, g$, or b represent red, green, and blue, respectively. This equation enables the recovery of an image by using only a portion of the image with N_p pixels from the original image with N pixels. It can be rewritten as

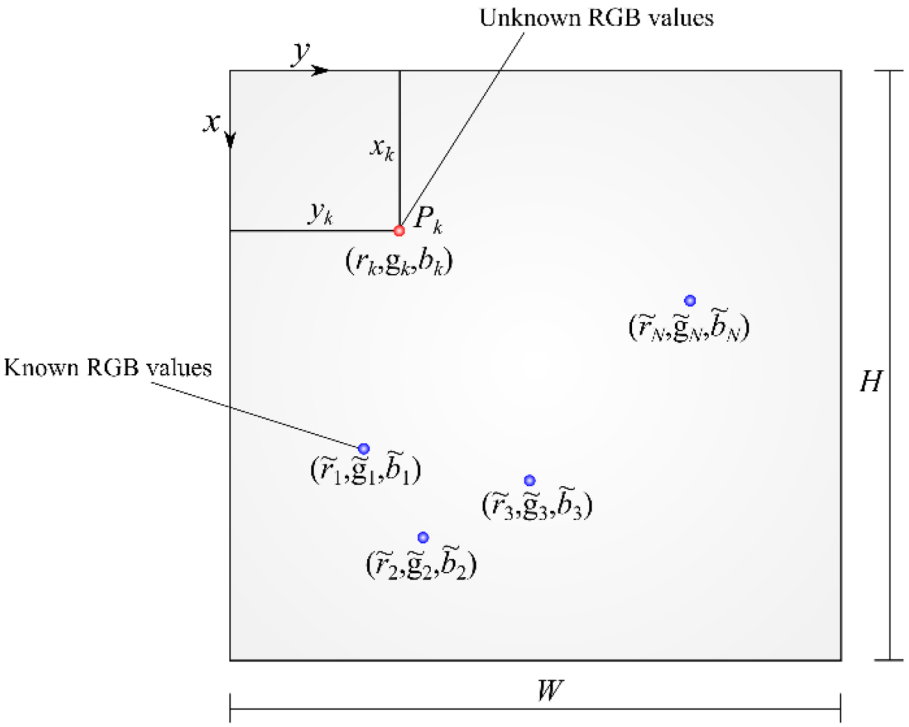


Fig. 5 Input and output points for PD image recovery

$$f_c(\mathbf{x}_k) = \sum_{j=1}^{N_p} H\left(\delta - \xi_{kj}\right)\omega_{kj} g_2^{00}\left(\xi_{kj}\right)\tilde{f}_c\left(\tilde{\mathbf{x}}_j\right)\tilde{A}_j \quad (k = 1, \ldots, N_u) \text{ for } c = r, g, b \tag{36}$$

where N_p and N_u indicate the number of picked and unpicked pixels, $H(\delta - \xi_{kj})$ is the Heaviside step function, and weight function ω_{kj} is defined as

$$\omega_{kj} = e^{-4\xi_{kj}^2/\delta_k^2} \tag{37}$$

Note that the total of the number of picked and unpicked pixels are equal to the total number of pixels of the original image, i.e., $N = N_p + N_u$.

The total area of the image, A, can expressed as

$$A = \sum_{j=1}^{N_p} A_j + \sum_{k=1}^{N_u} A_k \tag{38}$$

where A_j and A_k denote the areas of each pixels in the picked and unpicked portions of the image. Also, the total area of the image is distributed to the unknown lumped areas of each picked pixels, $\tilde{A}_j$, as

$$A = \sum_{j=1}^{N_p} \tilde{A}_j \tag{39}$$

Equating these expressions for the total area of the image leads

$$\sum_{j=1}^{N_p} \tilde{A}_j = \sum_{j=1}^{N_p} A_j + \sum_{k=1}^{N_u} A_k \tag{40}$$

By using Eqs. (14) and (15), the area of each unpicked pixel, A_k, can be expressed as

$$A_k = \sum_{m=1}^{N_p} A_{km} = \sum_{m=1}^{N_p} \rho_{km} A_k \tag{41}$$

Substituting from Eq. (41) into Eq. (40) yields

$$\sum_{j=1}^{N_p} \tilde{A}_j = \sum_{j=1}^{N_p} A_j + \sum_{k=1}^{N_u} \left(\sum_{m=1}^{N_p} \rho_{km} \right) A_k \tag{42}$$

or

$$\sum_{j=1}^{N_p} \left[\tilde{A}_j - \left(A_j + \sum_{k=1}^{N_u} \rho_{kj} A_k \right) \right] = 0 \tag{43}$$

Hence, the unknown lumped areas, $\tilde{A}_j$, associated with the picked pixels are determined as

$$\tilde{A}_j = A_j + \sum_{k=1}^{N_u} \rho_{kj} A_k \tag{44}$$

Noting that A_k and A_j are identical with $A_j = \Delta A$ and $A_k = \Delta A$ with ΔA being the area of each pixel in the original image, this equation can be rewritten as

$$\tilde{A}_j = \left(1 + \sum_{k=1}^{N_u} \rho_{kj} \right) \Delta A \tag{45}$$

Furthermore, the area of each pixel in the original image can be set to 1 for convenience, thus leading to

$$\tilde{A}_j = 1 + \sum_{k=1}^{N_u} \rho_{kj} \quad \text{for} \quad (j = 1, \ldots, N_p) \tag{46}$$

with

$$\rho_{kj} = \frac{1/\xi_{kj}^2}{\sum_{m=1}^{N_p} 1/\xi_{km}^2} \tag{47}$$

The recovered pixel values at the unpicked points are compared against the true image by comparing the predicted image colors to the original image colors. The convergence is reached if more than 90% of the original image is recovered. Otherwise, the analysis is repeated by picking more pixels from the original image.

The original image with N pixels for each color, $c = r, g$, or b, can be stored in a vector, $\mathbf{f}_c^*$, as

$$\mathbf{f}_c^* = \left\{ f_{c(1)}^*, f_{c(2)}^*, \cdots, f_{c(N)}^* \right\}^T \tag{48}$$

The iterative procedure adaptive image compression and recovery involves the following steps:

1. Start by picking a uniformly distributed pixels of about 1% of the total of N pixels from the original image and store them into the array of picked pixels, $\tilde{\mathbf{f}}_c$, as

$$\tilde{\mathbf{f}}_c^T = \{ f_{c(p_1)}^*, f_{c(p_2)}^*, \cdots, f_{c(p_{N_p})}^* \} \text{ for } c = r, g, b \tag{49}$$

   where p_k with $(k = 1, \cdots, N_p)$ represents the index IDs of the picked pixels and N_p is the number of picked pixels, which is initially close to $N_p \approx 0.01N$. The initial grid points are picked based on a coarse structured grid with uniform spacing because it covers the entire domain. Also, the unpicked pixels with unknown values are stored in vector, $\mathbf{f}_c$, containing

$$\mathbf{f}_c^T = \{ f_{c(u_1)}, f_{c(u_2)}, \cdots, f_{c(u_{N_u})} \} \text{ for } c = r, g, b \tag{50}$$

   where u_k with $(k = 1, \cdots, N_u)$ denotes the index ID numbers of the unpicked pixels with N_u being the number of unpicked pixels.
2. Use Eq. (36) along with Eq. (46) to predict the colors of unpicked pixels. Note that each color code has integer values and varies between 0 and 255. For this reason, the computed values of the unpicked pixels must be converted to the nearest integer between 0 and 255, i.e., $f_{c(u_k)} = \text{Round}(f_{c(u_k)})$, $f_{c(u_k)} = 0$ if $\text{Round}(f_{c(u_k)}) < 0$ and $f_{c(u_k)} = 255$ if $\text{Round}(f_{c(u_k)}) > 255$ for $k = 1, \cdots, N_u$;
3. Compute the difference between the color values of the original image and the unpicked pixels, whose values are predicted by Eq. (36) and rounded to integers in step 2 as

$$\Delta f_{c(u_k)} = \left| f_{c(u_k)}^* - f_{c(u_k)} \right| \quad (k = 1, \cdots, N_u) \tag{51}$$

4. Sort the difference values, $\Delta f_{c(u_k)}$, in descending order and store them in an array, Δf_c, defined as

$$\Delta \mathbf{f}_c^T = \left\{ \Delta f_{c(n_1)}, \Delta f_{c(n_2)}, \cdots, \Delta f_{c(n_{N_u})} \right\} \text{ for } c = r, g, b \tag{52}$$

   where the indices n_k with $(k = 1, \cdots, N_u)$ are such that

$$\text{Max}(\Delta f_{c(n_1)}) > \text{Max}(\Delta f_{c(n_2)}) > \cdots > \text{Max}(\Delta f_{c(n_{N_u})}) \tag{53}$$

   in which $\text{Max}(\Delta f_{c(n_k)})$ is defined as

$$\text{Max}(\Delta f_{c(n_k)}) = \text{Max}\left\{ \Delta f_{r(n_k)}, \Delta f_{b(n_k)}, \Delta f_{g(n_k)} \right\} \ (k = 1, \cdots, N_u) \tag{54}$$

5. Calculate the error, e, due to $\Delta f_{c(n_k)}$ defined in the form

$$e = \frac{\sum_{k=1}^{N_u} \left(\Delta f_{r(n_k)} + \Delta f_{b(n_k)} + \Delta f_{g(n_k)} \right)}{3 \times N \times 255} \tag{55}$$

6. If $e < 0.01$, convergence is achieved; print the converged image and stop the analysis. Otherwise, continue with step 7.
7. Pick additional M pixels with indices n_1 through n_M from the original image and append them to the vector of picked pixels, $\mathbf{f}_c$, in Eq. (49) as

$$\overset{\sim}{\mathbf{f}}_c^T = \left\{ f^*_{c(p_1)}, f^*_{c(p_2)}, \cdots, f^*_{c(p_{N_p})}, f^*_{c(n_1)}, f^*_{c(n_2)}, \cdots, f^*_{c(n_M)} \right\} \text{ for } c = r, g, b \tag{56}$$

   Limit the size of M not more than 5% of the total number of pixels, N. The indices for these additional pixels are chosen from the first M indices of vector $\Delta \mathbf{f}_c$ in Eq. (52) for faster convergence. Subsequently, remove the pixels with indices n_1 through n_M from the vector of unpicked pixels, in Eq. (50), to balance the total of picked and unpicked pixels, in the form

$$\mathbf{f}_c^T = \left\{ f_{c(n_{M+1})}, f_{c(n_{M+2})}, \cdots, f_{c(n_{N_u})} \right\} \tag{57}$$

   Also, update the number of picked and unpicked pixels as $N_p = N_p + M$ and $N_u = N_u - M$.
8. Continue performing steps 2 through 8 until convergence is reached in step 6.

6 Peridynamic Regression for Model Order Reduction

The PD regression can be employed to link two levels of discretization as shown in Fig. 6. The level-1 discretization is coarse and it enables reduction in the number of unknowns in the expression. The level-2 discretization is fine and controls the accuracy of evaluations. In the coarse grid shown in Fig. 6, the spacing between the green points is defined by $\Delta x_1 = L/(m-1)$ where m represents the number of points in $x-$ and $y-$ directions. It results in $M = m \times m$ points in the discretization of the domain. The position vector, the functional value, and the volume of j-th point in level-1 grid are designated as $\mathbf{x}_j$, w_j, and A_j, respectively.

In level-2 grid, the spacing between the blue points is defined by $\Delta x_2 = L/n$ where $N = n \times n$ represents the number of PD points. In this fine grid, the position vector, the functional value, and the area of k-th point are denoted by $\hat{x}_k$, $\hat{w}_k$, and $\hat{A}_k$, respectively. As shown in Fig. 6, the horizon (radius) of the k-th PD point in level-2 grid is denoted by $\hat{\delta}_k$. The distance between k - th PD point of level-1 grid and any other PD point in the level-2 grid is represented by $\xi_{kj} = x_j - \hat{x}_k$.

Using the PDDO introduced by Madenci et al. [8], the functional values of the points in the fine grid, $\hat{w}_{(k)}$, as well as their derivatives can be expressed in terms of the unknown functional values of the points in the coarse grid $w_{(j)}$ provided that the total area of the points in both grids is preserved as

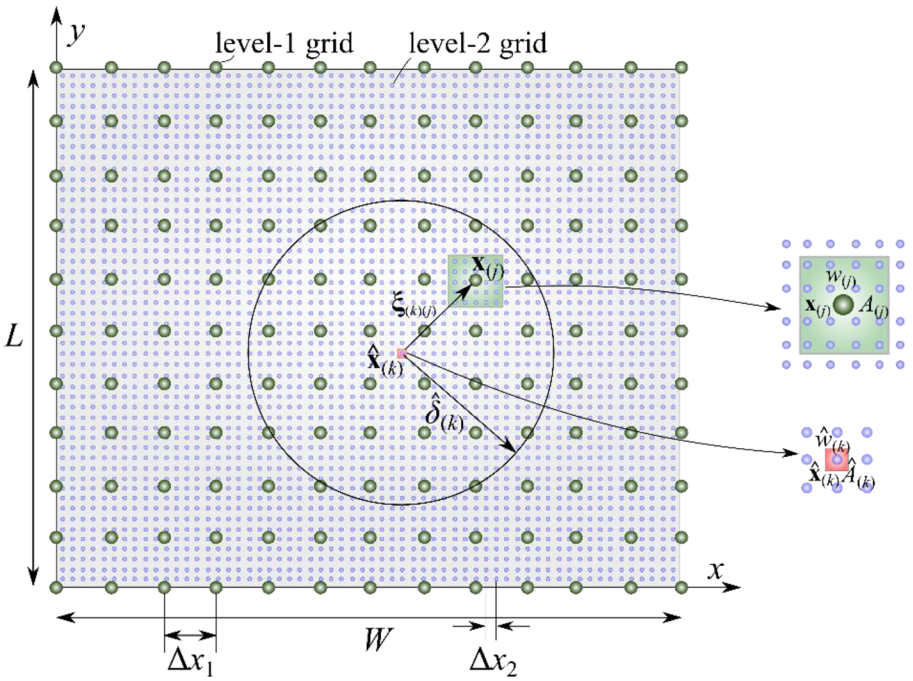


Fig. 6 Discretization of domain with level-1 (coarse) and level-2 (refined) grids

$$\sum_{j=1}^{M} A_j = \sum_{k=1}^{N} \hat{A}_k. \tag{58}$$

As shown in Fig. 6, the area of each PD point in the coarse grid, A_j, and fine grid, $\hat{A}_k$, can be calculated as

$$A_j = LW/M \quad j = 1, \ldots, M \tag{59}$$

and

$$\widehat{A}_k = LW/N \quad k = 1, \ldots, N. \tag{60}$$

Using the concept of PD regression, the functional value and its derivatives at each point in the fine grid can be expressed as

$$\hat{w}(\hat{\mathbf{x}}_k) = \sum_{j=1}^{N_k} w_j g_2^{00}(\xi_{kj};\omega_{kj}) A_j, \tag{61}$$

$$\hat{w}_{,x}(\hat{\mathbf{x}}_k) = \sum_{j=1}^{N_k} w_j g_2^{10}(\xi_{kj};\omega_{kj}) A_j, \tag{62}$$

$$\hat{w}_{,y}\left(\hat{\mathbf{x}}_k\right) = \sum_{j=1}^{N_k} w_j g_2^{01}\left(\xi_{kj};\omega_{kj}\right)A_j, \tag{63}$$

$$\hat{w}_{,xx}\left(\hat{\mathbf{x}}_k\right) = \sum_{j=1}^{N_k} w_j g_2^{20}\left(\xi_{kj};\omega_{kj}\right)A_j, \tag{64}$$

$$\hat{w}_{,xy}\left(\hat{\mathbf{x}}_k\right) = \sum_{j=1}^{N_k} w_j g_2^{11}\left(\xi_{kj};\omega_{kj}\right)A_j, \tag{65}$$

and

$$\hat{w}_{,yy}(\hat{\mathbf{x}}_k) = \sum_{j=1}^{N_k} w_j g_2^{02}(\xi_{kj};\omega_{kj})A_j \tag{66}$$

In these equations, N_k represents the number of level-1 points within the horizon of level-2 point,$\hat{\mathbf{x}}_{(k)}$. In matrix form, Eqs. (61)–(66) can be rewritten as

$$\hat{w}\left(\hat{\mathbf{x}}_k\right) = \mathbf{h}^T\left(\hat{\mathbf{x}}_k\right)\hat{\mathbf{w}}_k, \tag{67}$$

$$\hat{w}_{,x}\left(\hat{\mathbf{x}}_k\right) = \mathbf{h}_{,x}^T\left(\hat{\mathbf{x}}_k\right)\hat{\mathbf{w}}_k, \tag{68}$$

$$\hat{w}_{,y}\left(\hat{\mathbf{x}}_k\right) = \mathbf{h}_{,y}^T\left(\hat{\mathbf{x}}_k\right)\hat{\mathbf{w}}_k, \tag{69}$$

$$\hat{w}_{,xx}\left(\hat{\mathbf{x}}_k\right) = \mathbf{h}_{,xx}^T\left(\hat{\mathbf{x}}_k\right)\hat{\mathbf{w}}_k, \tag{70}$$

$$\hat{w}_{,xy}\left(\hat{\mathbf{x}}_k\right) = \mathbf{h}_{,xy}^T\left(\hat{\mathbf{x}}_k\right)\hat{\mathbf{w}}_k, \tag{71}$$

and

$$\hat{w}_{,yy}(\hat{\mathbf{x}}_k) = \mathbf{h}_{,yy}^T(\hat{\mathbf{x}}_k)\hat{\mathbf{w}}_k \tag{72}$$

where the coefficient vectors, $\mathbf{h}(\hat{\mathbf{x}}_k)$, $\mathbf{h}_{,x}(\hat{\mathbf{x}}_k)$, $\mathbf{h}_{,y}(\hat{\mathbf{x}}_k)$,$\mathbf{h}_{,xx}(\hat{\mathbf{x}}_k)$, and $\mathbf{h}_{,yy}(\hat{\mathbf{x}}_k)$, and the unknown vector, $\hat{\mathbf{w}}_k$, are defined as

$$\mathbf{h}^T\left(\hat{\mathbf{x}}_k\right) = \left\{g_2^{00}\left(\xi_{k1};\omega_{k1}\right)A_1, .., g_2^{00}\left(\xi_{kN_k};\omega_{kN_k}\right)A_{N_k}\right\}, \tag{73}$$

$$\mathbf{h}_{,x}^T\left(\hat{\mathbf{x}}_k\right) = \left\{g_2^{10}\left(\xi_{k1};\omega_{k1}\right)A_1, .., g_2^{10}\left(\xi_{kN_k};\omega_{kN_k}\right)A_{N_k}\right\}, \tag{74}$$

$$\mathbf{h}_{,y}^T\left(\hat{\mathbf{x}}_k\right) = \left\{g_2^{01}\left(\xi_{k1};\omega_{k1}\right)A_1, .., g_2^{01}\left(\xi_{kN_k};\omega_{kN_k}\right)A_{N_k}\right\}, \tag{75}$$

$$\mathbf{h}_{,xx}^T\left(\hat{\mathbf{x}}_k\right) = \left\{g_2^{20}\left(\xi_{k1};\omega_{k1}\right)A_1, .., g_2^{20}\left(\xi_{kN_k};\omega_{kN_k}\right)A_{N_k}\right\}, \tag{76}$$

$$\mathbf{h}^T_{,xy}(\hat{\mathbf{x}}_k) = \left\{ g_2^{11}(\xi_{k1};\omega_{k1})A_1, .., g_2^{11}(\xi_{kN_k};\omega_{kN_k})A_{N_k} \right\}, \tag{77}$$

$$\mathbf{h}^T_{,yy}(\hat{\mathbf{x}}_k) = \left\{ g_2^{02}(\xi_{k1};\omega_{k1})A_1, .., g_2^{02}(\xi_{kN_k};\omega_{kN_k})A_{N_k} \right\} \tag{78}$$

and

$$\hat{\mathbf{w}}^T_k = \left\{ w_1, w_2, \dots, w_{N_k} \right\}. \tag{79}$$

The major implication of Eq. (67) is that the zeroth-order PDDO can be used for interpolation. Hence, the PD regression for model order reduction can be used to approximate the field variables such as the displacements of a point in the material based on the principle of minimum potential energy. The present approach can readily be used in the solution of structural problems where the equilibrium equations are derived based on energy principles as demonstrated by Madenci et al. [9, 10].

7 Numerical Results

The numerical results concern interpolation of real data in two and three dimensions, interpolation to approximate a three-dimensional function, adaptive data recovery in three-dimensional space, recovery of missing pixels in an image, adaptive image compression and recovery, and free energy function of a thermally fluctuating rod through model reduction.

7.1 Interpolation of Temperature Data

The data consists of the maximum temperature readings across 121 weather stations with the corresponding latitude, longitude, and elevation of each weather station. This weather data on January 4, 2020 in Arizona is obtained from NOAA [11]. The exact locations and temperature readings are given in Appendix 3. In order to demonstrate the capability of the present approach, 10 data points shown in Fig. 7 as red crosses are removed at random from the original 121 data points. The remaining 111 data points serve as input points. The locations and temperature readings of the randomly removed stations are shown in Table 1. Figure 8 shows the PD interpolation grid overlayed on the Arizona map. The family population, N_k, of each output point, $\mathbf{x}_k$, is defined by including input points within its horizon, δ. Therefore, the number of family members may be different for each output point. The family member selection is achieved using a k-d tree nearest neighbor algorithm [4].

The temperature is estimated at each output point on the grid. The values at the output points are estimated while recovering the original 111 data points for both two- and three-dimensional interpolation. The PD interpolation is performed through Eq. (23) along with Eq. (20) for $p = 2$, and the PD functions are constructed by truncating the TSE after the first-order derivatives. The PD temperature predictions at the output points closest to the red circles are compared with their original readings in order to measure the error.

In the case of a two-dimensional data interpolation, temperature, T, varies over x and y representing the longitude and latitude, respectively. The number of output points is 40×40, and they are equally spaced in the region of $-115^\circ \le x \le -109.1^\circ$ and $-31^\circ \le \mathrm{y} \le 37.5^\circ$. The grid spacing is defined by $\Delta x = 0.1475^\circ$ and $\Delta y = 0.1525^\circ$ in the

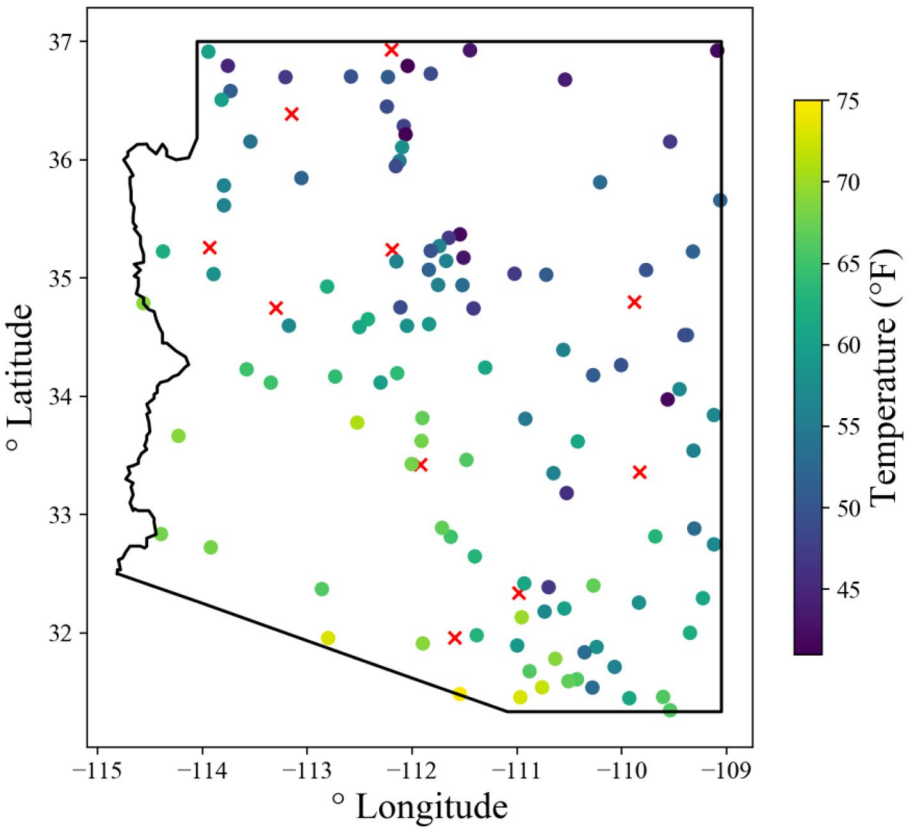


Fig. 7 Location of input and output (red) points given in Table 1

longitudinal and latitude directions, respectively. The family population of each output point, N_k, is defined by including input points within a radius of 3°. The horizon, δ_k, is determined by the furthest input point from $\mathbf{x}_k$.

Table 1 Locations and values of data points removed from the original data set

Weather station	Elevation (m)	Latitude	Longitude	Temp. (°F)
Williams	2105.9	35.2413	−112.1929	56
Tohono Chul	770.2	32.3391	−110.9808	63
Nixon Flats	1981.2	36.39	−113.1522	54
Buckskin Mountain	1950.7	36.9306	−112.1997	48
Dry Lake	2264.1	33.3597	−109.8331	60
Petrified Forest	1659.9	34.7994	−109.885	49
Goodwin Mesa	1280.2	34.75	−113.3	63
Kitt Peak	2069.6	31.9602	−111.5979	60
Kingman Airport	1042.4	35.2577	−113.933	63
Tempe ASU	355.7	33.4258	−111.9216	71

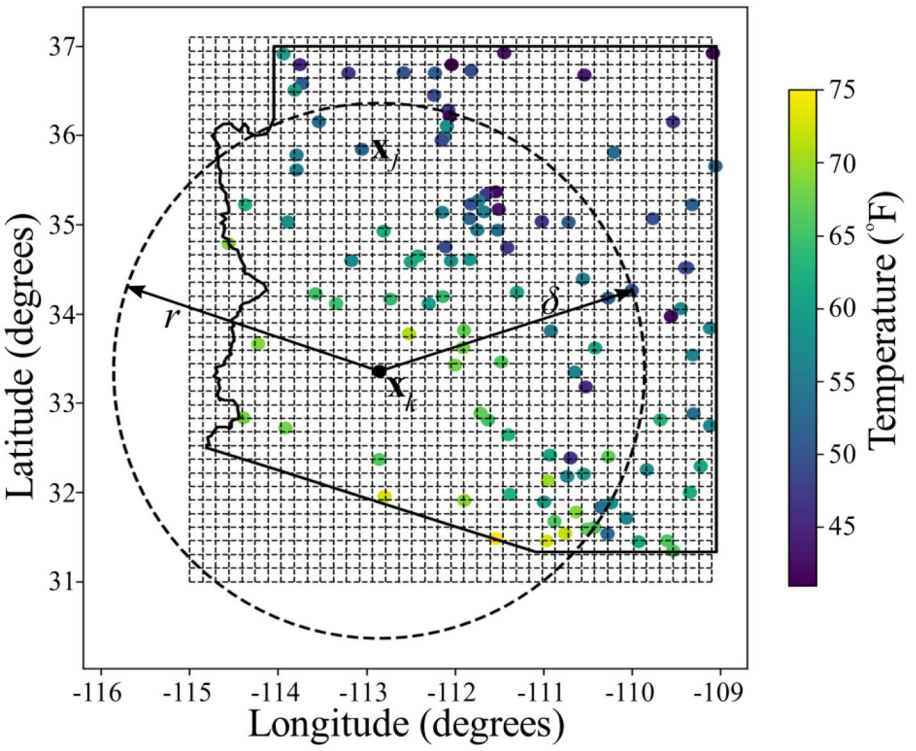


Fig. 8 PD grid of output points overlayed on the Arizona map and the horizon of point $\mathbf{x}_k$

Figure 9 shows the PD interpolation values at the output points. The values with closest coordinates to the red circles are shown in Table 2. These PD estimates are compared with their original values and the predictions based on Ordinary Kriging method [4]. The Kriging algorithm described in Appendix 2 is implemented using the PyKrige python software package. The error, e, between the readings and predictions is calculated as 4.54 and 5.01 for PD and Kriging estimations, respectively.

In order to demonstrate the performance of PD interpolation in a three-dimensional space, the elevation is also included in the input data. In the case of a three-dimensional data, temperature, T varies over x, y, and z representing the longitude, latitude, and elevation, respectively. The number of output points is $25 \times 25 \times 30$, and they are equally spaced in the region of $(-115^\circ \leq x \leq -109.1^\circ)$, $(-31^\circ \leq y \leq 37.5^\circ)$, and $(0 \leq z \leq 3000$ m). The grid spacing is defined by $\Delta x = 0.236^\circ$, $\Delta y = 0.244^\circ$, and $\Delta z = 100$ m in the longitudinal, latitude, and elevation directions, respectively. The family population of each unknown point, N_k, is defined by including input data points within a cylindrical interaction domain. Its radius is defined as $\delta = 1.9^\circ$ with a height of 3000 m. As shown in Fig. 10, the horizon, δ_k, is determined by the furthest input point from the output point, $\mathbf{x}_k$.

The PD interpolation values at the output points with closest coordinates to the red circles are shown in Table 3. These PD estimates are compared with their original values and the predictions based on the Ordinary Kriging method [4]. The error is calculated as 5.14

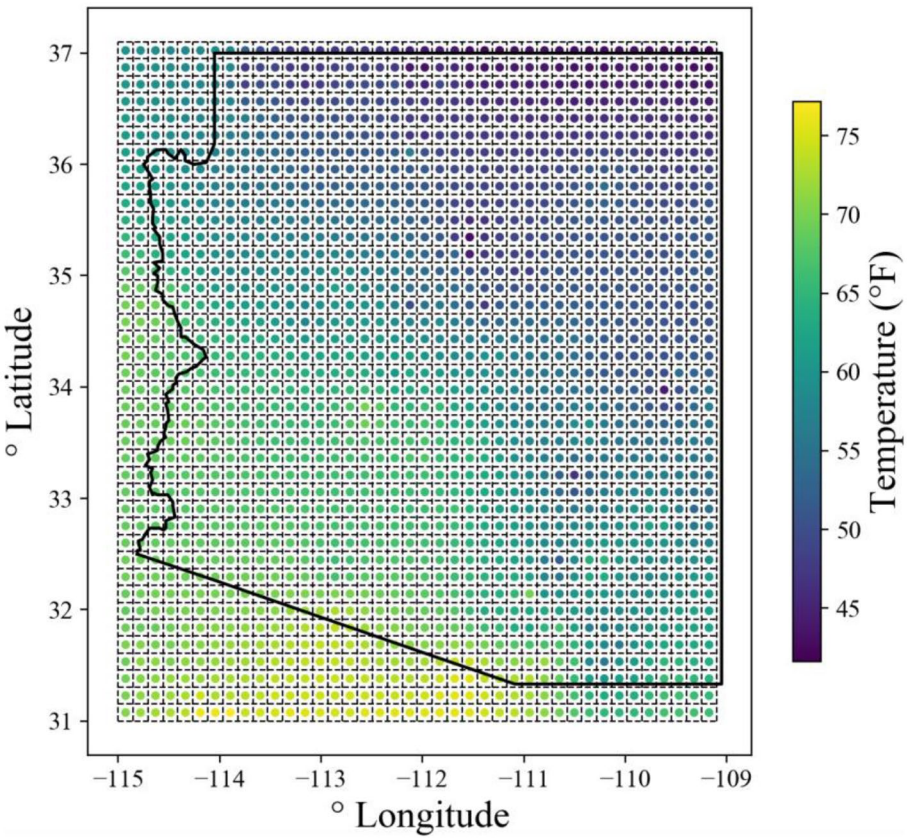


Fig. 9 PD interpolation values on the Arizona map

and 6.72 for PD and Kriging estimations, respectively. The 3D interpolation is not as robust as the 2D interpolation due to the sparsity of the data. Introducing another dimension without increasing the number of data points suffers from the curse of dimensionality.

Table 2 PD and Kriging estimates of the temperature based on the 2D coordinates of weather stations

Weather station	Original temp	PD interpolation	Kriging
Williams	56	55.2064	54.1327
Tohono Chul	63	62.155	61.1788
Nixon Flats	54	50.6628	51.1145
Buckskin Mountain	48	47.2551	46.7098
Dry Lake	60	55.7386	54.9305
Petrified Forest	49	51.1166	50.6425
Goodwin Mesa	63	59.0222	60.1946
Kitt Peak	60	65.6535	67.4892
Kingman Airport	63	58.3842	59.3514
Tempe ASU	71	67.6238	66.6457

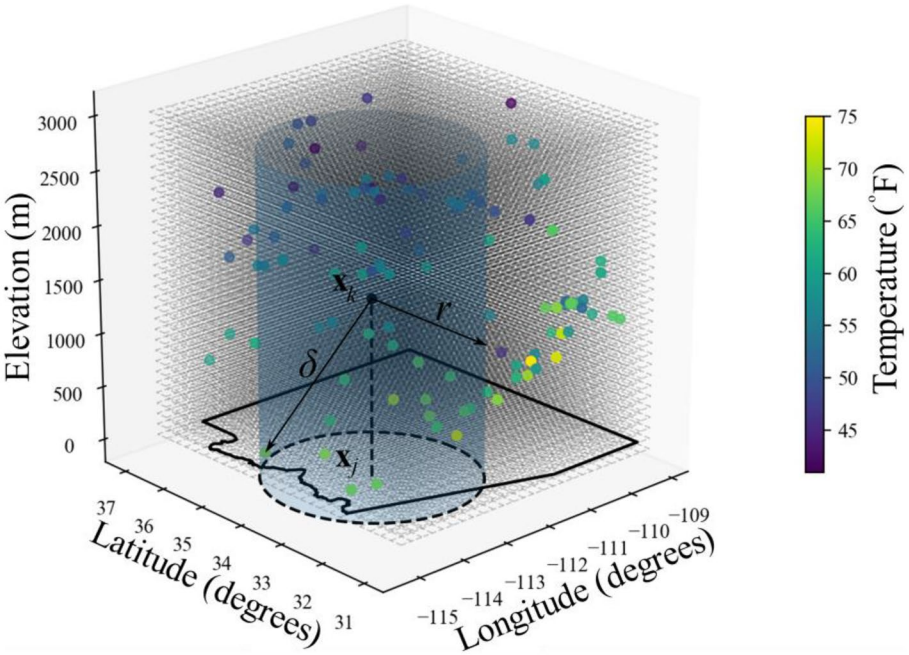


Fig. 10 Family members and horizon of point $\mathbf{x}_k$ encompassing the input points in a cylindrical interaction domain

7.2 Interpolation to Approximate a Function

The input data shown in Fig. 11 is generated randomly at 270 points within a unit cube $(0 \le x, y, z \le 1)$ by evaluating the following function:

$$f(x, y, z) = [1 - \sin(\pi x)]y^3 e^z \tag{80}$$

The number of output points is $30 \times 30 \times 30$, and they are equally spaced with grid spacing by $\Delta x = \Delta y = \Delta z = 1/30$. The randomly generated input data constitutes 1% of the PD grid points. The PD interpolation is performed at the 27,000 output points through

Table 3 PD and Kriging estimates of the temperature based on the 3D coordinates of weather stations

Weather station	Original temp	PD interpolation	Kriging
Williams	56	56.3091	51.5217
Tohono Chul	63	57.6099	61.6683
Nixon Flats	54	50.5367	49.499
Buckskin Mountain	48	45.1787	50.5259
Dry Lake	60	53.3049	50.871
Petrified Forest	49	52.3342	53.2365
Goodwin Mesa	63	59.135	59.7036
Kitt Peak	60	63.864	51.177
Kingman Airport	63	61.7043	64.434
Tempe ASU	71	67.5256	67.2911

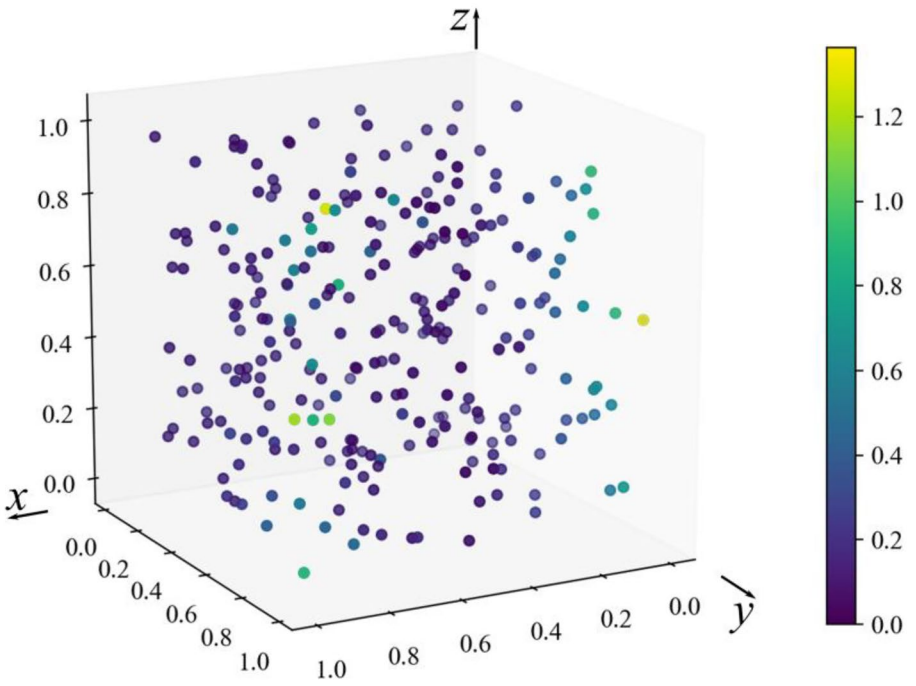


Fig. 11 Randomly generated input data

Eq. (23) along with Eq. (21) for $p = 3$. The family population of each unknown point, N_k, is defined by the closest 50 PD points. Also, the PD functions are constructed by truncating the TSE after the third-order derivatives, i.e., $N = 3$. The PD estimates of the function value at the output points are shown in Fig. 12. The error measure, e, between the functional value and the estimates is 0.2943%.

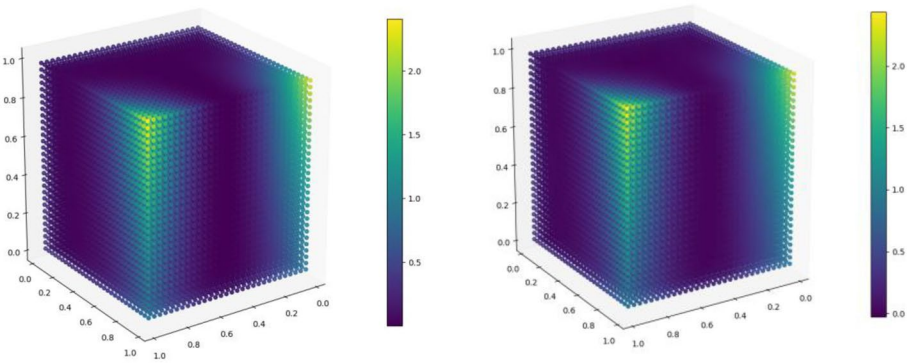


Fig. 12 Functional variation at the output points: (**a**) actual data and (**b**) PD recovery

The PD recovery of data is compared against the actual data along the line specified by $x = y = z$ in Figs. 13 and 14. Also, these figures show the number of picked input data points along this line.

7.3 Adaptive Data Recovery

In order to demonstrate data recovery, the data is fabricated by using the following function:

$$f^*(x, y, z) = \frac{2xyz}{x^2 + y^2 + z^2} \quad \text{for} \quad -1 \le x, y, z \le 1 \quad -1 \le x, y, z \le 1 \tag{81}$$

The data is generated for a grid spacing of $40 \times 40 \times 40$ in the 3-D space. The PD regression at each unknown point is performed through Eq. (23) along with Eq. (21) for $p = 3$. The family population of each unknown point, N_k, is defined by a minimum of 50 pixels encompassed by a circle. As explained in Sect. 4, the initial data is 1% of the complete data and randomly picked, and subsequently the data is increased adaptively in 5% increments.

As shown in Fig. 15, the recovered data with the randomly picked initial data has an overall error of 3.745%. Although this error is high, the recovered data provides crucial information about the location of high gradients for the selection of additional 5% data points for the next iteration. Figure 16a shows the picked data points for the second

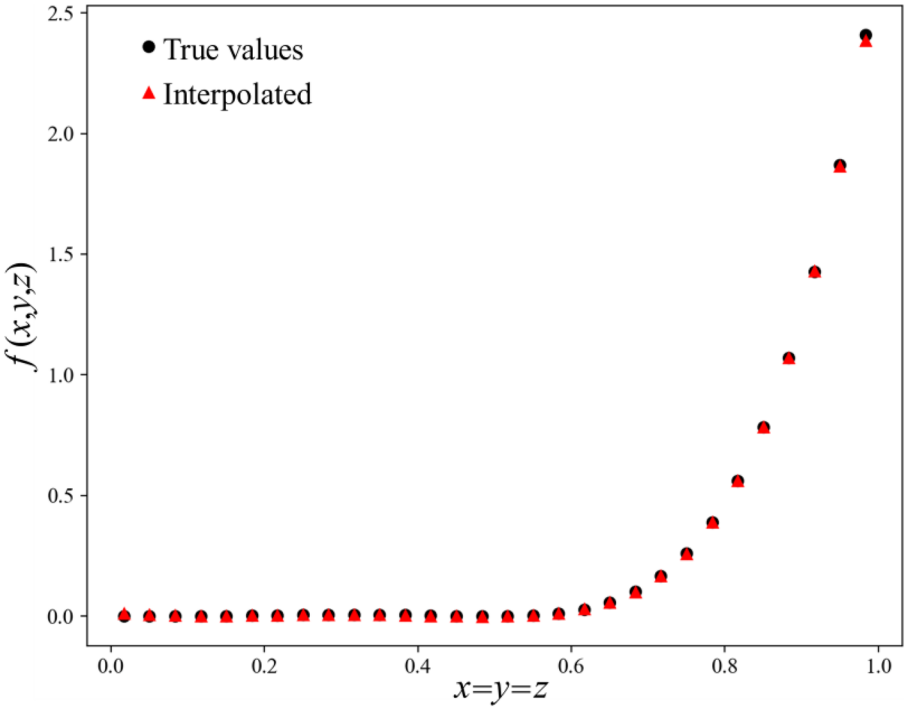


Fig. 13 Comparison of the actual data with PD recovery along the line ($x = y = z$)

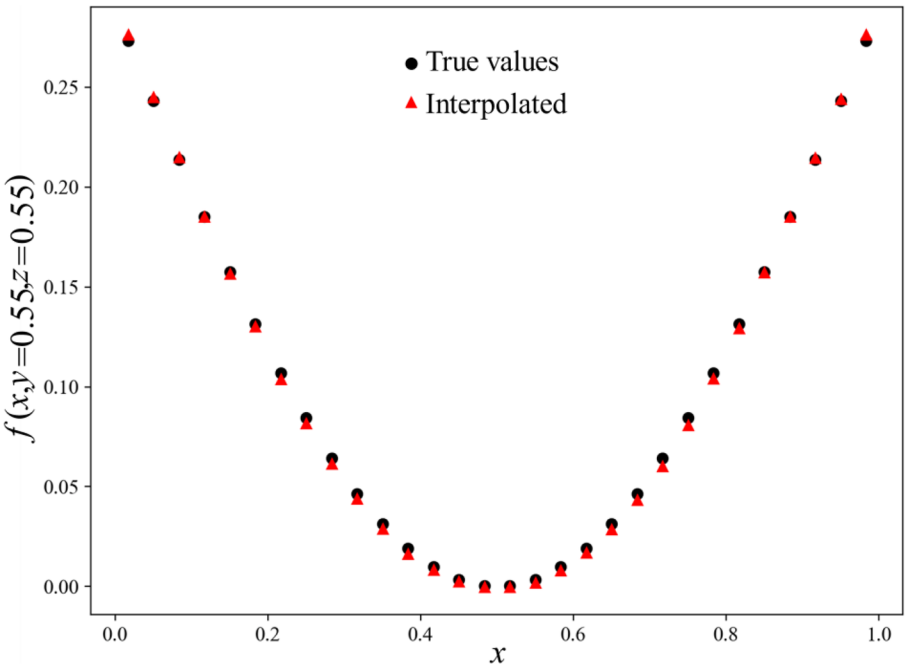


Fig. 14 Comparison of the actual data with PD recovery along the line $(x, y = 0.55, z = 0.55)$

iteration. The recovered data with only 6% of the total data points has overall error of 2% as shown in Fig. 16b. Finally, the adaptive selection of data points after the third iteration with only 11% of the total data points, shown in Fig. 17a, results in the desired error less than 1% against the original data set as shown in Fig. 17b. The PD regression successfully estimates the unknown functional values.

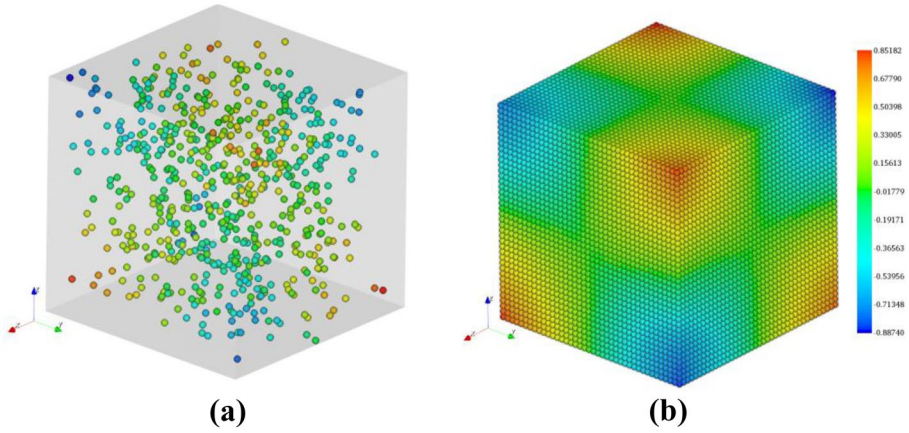


Fig. 15 Randomly picked 1% of the data population and recovered data with percentage error of 3.745

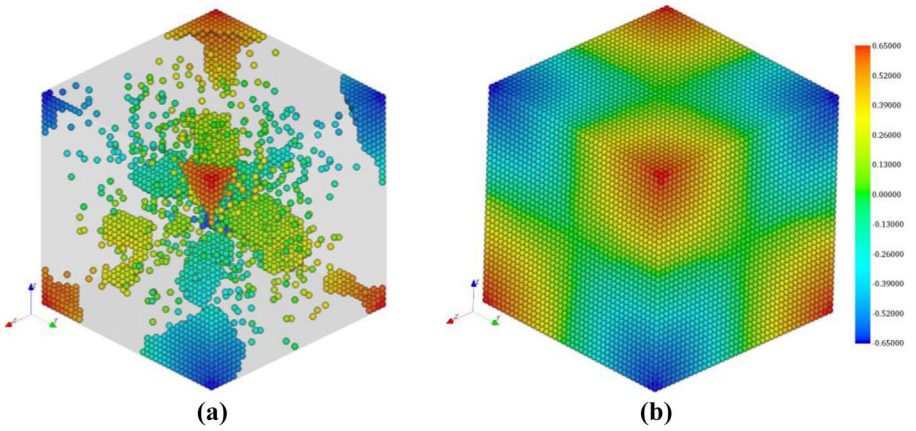


Fig. 16 Adaptively picked 6% of the data population and recovered data with percentage error of 2.0

7.4 Image Recovery

The image shown in Fig. 18 is constructed by 512×512 number of pixels specifying its resolution. The blue spots in Fig. 19a are randomly distributed and indicate the pixels of unknown values. The areas of known pixels are determined by using Eq. (46) for $p = 3$, and the PD regression at each unknown pixel is performed through Eq. (36). The family population of each unknown point, N_k, is defined by a minimum of 50 pixels encompassed by a circle.

The image with recovered pixels is shown in Fig. 19b. The PD interpolation successfully estimates the missing pixel values. The small spots of missing pixels have continuous color variations. The blue spots on the eyes are successfully recovered. However, the large spots of missing locations have some smeared color discontinuity. The overall error measured against the original image is 0.0256%.

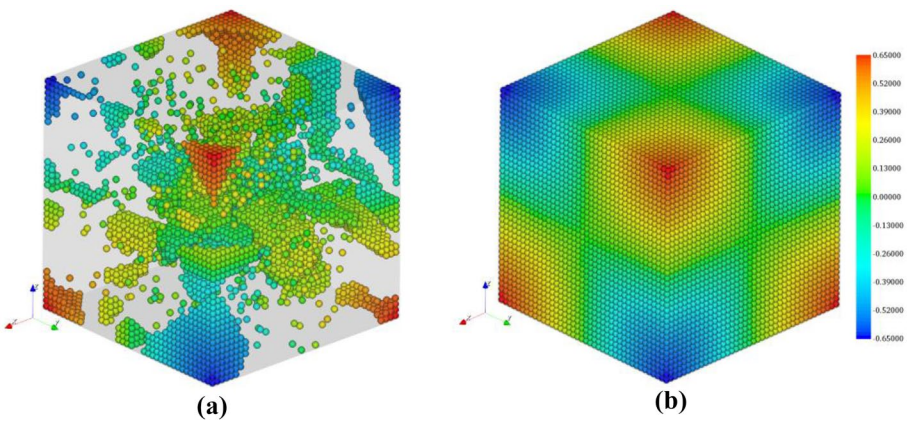


Fig. 17 Adaptively picked 11% of the data population and recovered data with percentage error of 0.847

Fig. 18 Image with complete pixels of $512 \times 512 = 262144$ [12]

7.5 Adaptive Image Compression

The adaptive data compression is applied to the image shown in Fig. 18. The areas of known pixels are determined by using Eq. (46) for $p = 3$, and the PD regression at each unknown pixel is performed through Eq. (36). The family population of each unknown point, N_k, is defined by a minimum of 50 pixels encompassed by a circle.

Since the critical pixels in the image are not known a priori, the initial set of 572 pixels (0.195313% of total pixels 262,144) are selected uniformly as shown in Fig. 20a. The

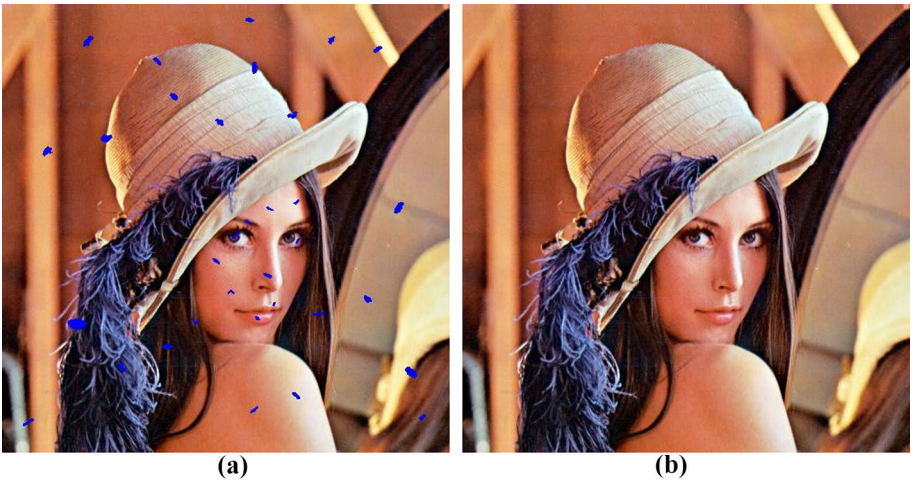


Fig. 19 Image with (**a**) missing pixels (blue spots) and (**b**) recovered pixels (error of 0.0256%)

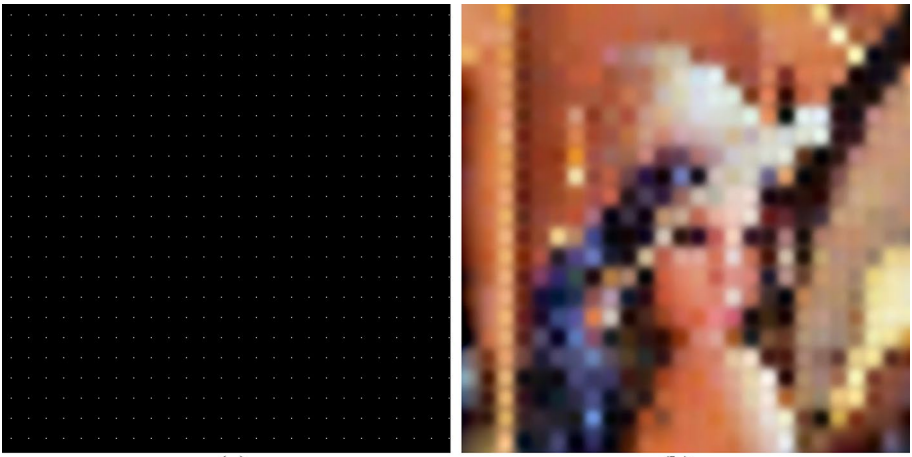


Fig. 20 Iteration 1 — images with (**a**) 572 picked pixels indicated with white (0.195313% of total pixels) and (**b**) recovered pixels (error of 9.067%)

image with recovered pixels is shown in Fig. 20b. Although the recovered image looks poor, it is obtained by using only 0.195313% of the total pixels of 262,144. The overall error is 9.067%. However, this image provides crucial information about the high gradients of color changes.

The new set of picked pixels is used in the next iteration with a 5% increase in pixel numbers. Figures 21, 22, 23, 24, 25, 26, 27, 28 and 29 show the adaptively picked pixels and recovered images for the next 9 iterations. As the number of picked pixels increases, the recovered image looks much improved, and the overall error reduces to 4.69422% with 15.21568% of total pixels at iteration 4 as shown in Fig. 23. The error reduces to

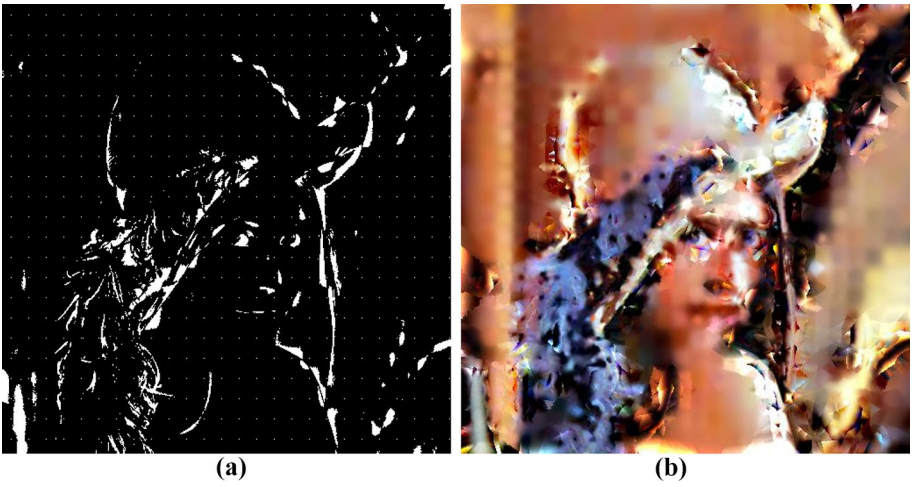


Fig. 21 Iteration 2 — images with (**a**) 13,677 picked pixels indicated with white (5.217361% of total pixels) and (**b**) recovered pixels (error of 13.3117%)

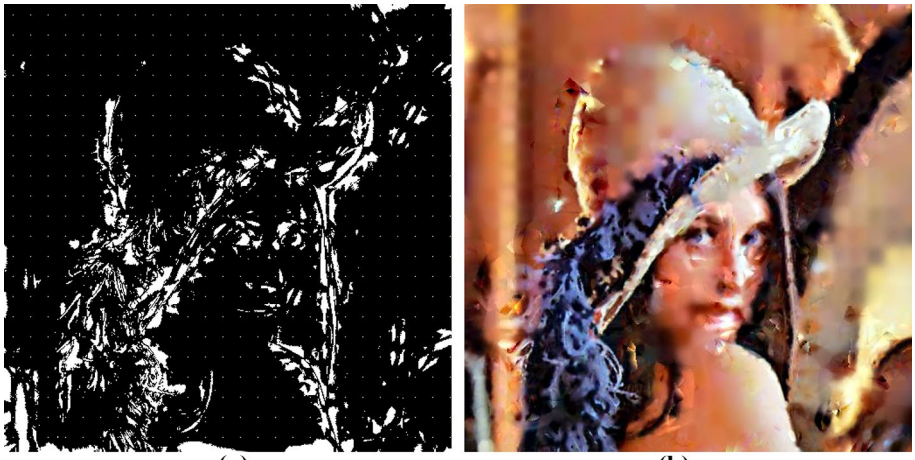


Fig. 22 Iteration 3 — images with (**a**) 26,782 picked pixels indicated with white (10.21652% of total pixels) and (**b**) recovered pixels (error of 6.85677%)

1.70792% with 30.21355% of total pixels as shown in Fig. 25 at iteration 6. Finally, the adaptive selection with 45.21103% of total pixels, shown in Fig. 29, results in a desired error (0.88737%) less than 1% against the original image as shown in Fig. 18. The original and the recovered images are shown in Fig. 30.

7.6 Model Reduction for Thermal Fluctuation of a Rod

The free energy function, $\mathcal{G}$, can be expressed in terms of the partition function as

$$\mathcal{G} = -k_B T \ln Z \tag{82}$$

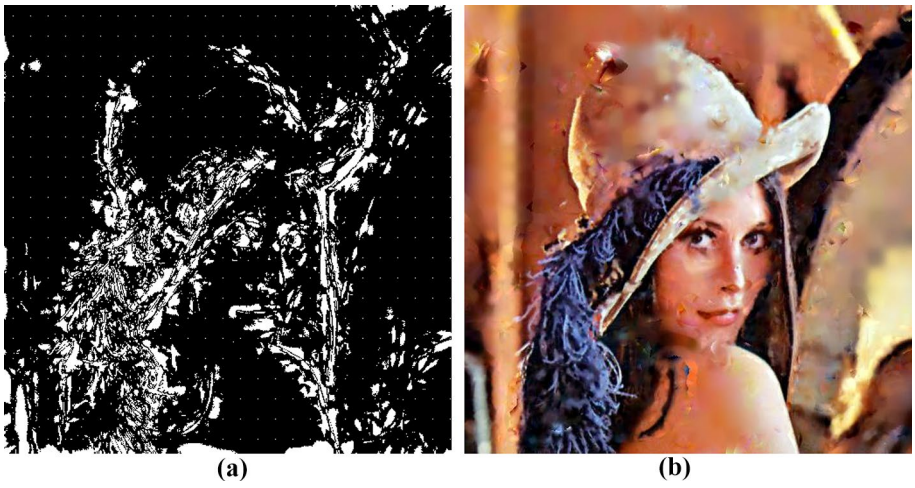


Fig. 23 Iteration 4 — images with (**a**) 39,887 picked pixels indicated with white (15.21568% of total pixels) and (**b**) recovered pixels (error of 4.69422%)

Fig. 24 Iteration 5 — images with (**a**) 52,992 picked pixels indicated with white (20.21484% of total pixels) and (**b**) recovered pixels (error of 3.23066%)

in which $k_{\mathrm{B}} = 1.38(\mathrm{pN})(\mathrm{nm})^{\circ}\mathrm{K}^{-1}$ is the Boltzmann constant, T is the temperature in $^{\circ}K$, and Z is the partition function. It can be expressed in the form of a multi-dimensional Gaussian integration as [13]

$$Z = \int e^{-U(w)/k_B T} d\mathbf{w}. \tag{83}$$

This integration can be carried out for the stored energy of a quadratic form as [14, 15]

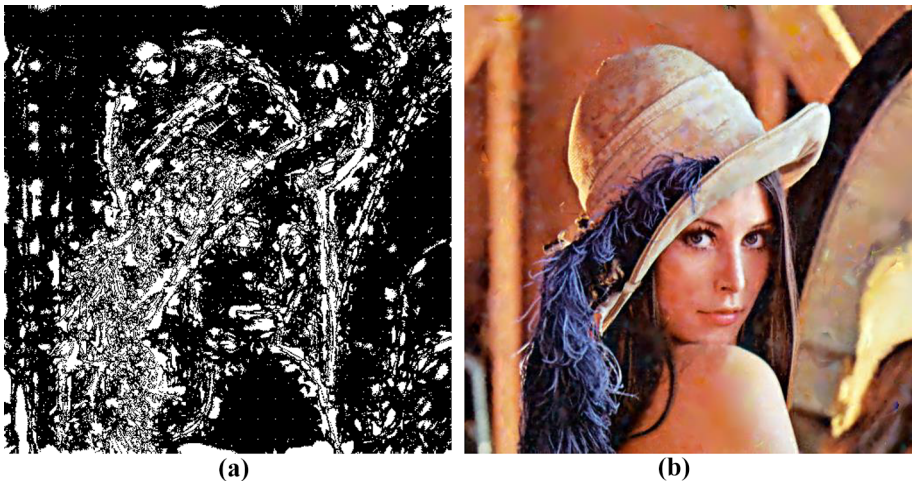


Fig. 25 Iteration 6 — images with (**a**) 66,098 picked pixels indicated with white (25.21439% of total pixels) and (**b**) recovered pixels (error of 2.33139%)

Fig. 26 Iteration 7 — images with (**a**) 79,203 picked pixels indicated with white (30.21355% of total pixels) and (**b**) recovered pixels (error of 1.70792%)

$$Z = \int e^{-U(w)/k_B T} d\mathbf{w} = e^{-\frac{U_{\min}}{k_B T}} \sqrt{\frac{\left(\pi k_B T\right)^M}{\det \mathbf{K}}} \tag{84}$$

where M is the number of unknowns in the vector of $\mathbf{w}$ which represents the possible deformation states. It fluctuates about the lowest elastic energy state $U_{\min}$ for $T > 0$. The symmetric and positive definite matrix, $\mathbf{K}$ of size $(M \times M)$, represents the stiffness of the structure. Substituting for the partition function, Z, and expanding the terms, the free energy function can be evaluated as

Fig. 27 Iteration 8 — images with (**a**) 92,308 picked pixels indicated with white (35.21271% of total pixels) and (**b**) recovered pixels (error of 1.3637%)

(a) **(b)**

Fig. 28 Iteration 9 — images with (**a**) 105,413 picked pixels indicated with white (40.21187% of total pixels) and (**b**) recovered pixels (error of 1.08388%)

$$\mathcal{G} = U_{\min} - \frac{k_B MT}{2}\ln(\pi k_B T) + \frac{k_B T}{2}[\ln(\det \mathbf{K})] \tag{85}$$

For a flat configuration with $U_{\min} = 0$, the free energy simplifies to

$$\mathcal{G} = -\frac{k_B TM}{2}\ln(\pi k_B T) + \frac{k_B T}{2}\left[\ln D_1 + \ln D_2 + \cdots + \ln D_M\right] \tag{86}$$

where D_i $(i = 1, \ldots, M)$ are the diagonal entries of the diagonal matrix $\mathbf{D}$ appearing in the decomposition as

(a) **(b)**

Fig. 29 Iteration 10 — images with (**a**) 118,518 picked pixels indicated with white (45.21103% of total pixels) and (**b**) recovered pixels (error of 0.88737%)

(a) **(b)**

Fig. 30 Images with (**a**) original and (**b**) recovered pixels with 45.21103% of total pixels

$$\mathbf{K} = \mathbf{L}\mathbf{D}\mathbf{L}^T \tag{87}$$

where $\mathbf{L}$ is a lower triangular matrix whose diagonals are equal to unity with $\det \mathbf{L} = 1$ and $\mathbf{D}$ has positive entries.

The evaluation of the free energy requires the computation of the determinant of $\mathbf{K}$. However, computing the determinant of an extremely large stiffness matrix poses computational challenges. Accurate evaluation of the determinant of the stiffness matrix is a key step in the calculation of the partition function. The simultaneous use of fine and coarse grids along with PD interpolation eliminates the computational challenges.

Su and Purohit [12] modeled the thermal fluctuation of a rod by considering its total energy, U, in the form

$$U = \frac{1}{2}\int_L K_B w_{,xx}^2 dx + \frac{1}{2}\int_L F w_{,x}^2 dx. \tag{88}$$

where w is the out-of-plane deflection and F represents the force. Its bending modulus, K_B, can be measured experimentally. The length of the rod is L. They derived the analytical expression for the reduction in length of a fluctuating rod as

$$\Delta L = \frac{1}{4}k_B T\left(\frac{L}{\sqrt{K_B F}}\coth\frac{FL}{\sqrt{K_B F}} - \frac{1}{F}\right) \tag{89}$$

By using Eq. (59), the reduction in length can be also expressed as

$$\Delta L = \frac{\partial \mathcal{G}}{\partial F} = \frac{k_B T}{2}\frac{d}{dF}\ln(\det \mathbf{K}) \tag{90}$$

The energy of each point $\hat{\mathbf{x}}_{(k)}$ in the fine grid can be expressed in the form

$$U_{(k)} = \frac{1}{2}K_B\left(\hat{w}_{(k),xx}\right)^2\hat{\ell}_{(k)} + \frac{1}{2}F\left(\hat{w}_{(k),x}^2\right)\hat{\ell}_{(k)} \tag{91}$$

with $\hat{\ell}_{(k)} = L/N$ representing the length of each point. After expanding this equation and substituting for the derivatives of $\hat{w}_{(k)}$ from Eqs. (68) and (70), the strain energy at point $\hat{\mathbf{x}}_{(k)}$ is expressed in matrix form as

$$U_{(k)} = \frac{1}{2}\hat{\mathbf{w}}_{(k)}^T \mathbf{K}_{(k)} \hat{\mathbf{w}}_{(k)} \tag{92}$$

where the stiffness matrix, $\mathbf{K}_{(k)}$, is defined as

$$\mathbf{K}_{(k)} = \left[K_B\left(\mathbf{h}_{(k),xx}\mathbf{h}_{(k),xx}^T\right) + F\left(\mathbf{h}_{(k),x}\mathbf{h}_{(k),x}^T\right)\right]\hat{\ell}_{(k)} \tag{93}$$

The total energy in the rod can be expressed in the form

$$U = \sum_{j=1}^{M} U_{(j)} = \frac{1}{2}\sum_{j=1}^{M} \hat{\mathbf{w}}_{(j)}^T \mathbf{K}_{(j)} \hat{\mathbf{w}}_{(j)} = \mathbf{w}^T \mathbf{K} \mathbf{w} \tag{94}$$

The symmetric and positive definite matrix, $\mathbf{K}$, of size $(M \times M)$ can be determined as

$$\mathbf{K} = \text{Assemble}\left[\mathbf{K}_{(1)}, \mathbf{K}_{(2)}, \cdots, \mathbf{K}_{(M)}\right], \tag{95}$$

The PD model of a rod is subjected to a force, F, varying from 200 to 2000 pN at a specified temperature of $T = 300$ °K. The contour length of the rod is $L = 2.5$ nm, and its flexural rigidity is specified as $K_B = 2.5 k_B T$ nm. Its persistence length, $p = K_B/(k_B T)$, is the same as the contour length.

The determinant of the rod stiffness matrix is computed for three different level-1 grid sizes specified by $m = 51$, $m = 101$, and $m = 201$ points. The corresponding level-2 grid division is achieved by $n = 4(m-1)$ points. The horizon of point, $\hat{\mathbf{x}}_{(k)}$, in the refined grid is specified as $\hat{\delta}_{(k)} = 5\Delta x_1$.

The PD evaluation of reduction in length, ΔL, is evaluated for specified force values of $F - \Delta F$, F, and $F + \Delta F$ with ΔF representing the incremental values for numerical differentiation based on central difference approximation as

$$\Delta L(F, T) = \frac{k_B T}{4\Delta F} \ln \frac{(\det \mathbf{K}(F + \Delta F, T))}{(\det \mathbf{K}(F - \Delta F, T))} \tag{96}$$

The extension in length defined as $(L - \Delta L)$ is computed for each discretization at $T = 300$ °K.

As shown in Fig. 31, the PD prediction provides sufficiently accurate results with 51 unknowns and converges to the analytical formula (the worm-like-chain relation for force extension), Eq. (96) with 201 unknowns. Su and Purohit [12] performed their analysis with 500 elements and recovered the analytical solution with 50,000 elements. The simultaneous use of fine and coarse grids along with PD interpolations reduces the number of unknowns and yet provides very accurate results. This reduction in degrees of freedom is an essential gain in capability especially for the computation of the determinant of stiffness matrix. It was also successfully applied in the evaluation of the partition function for a lipid membrane with inclusions [14].

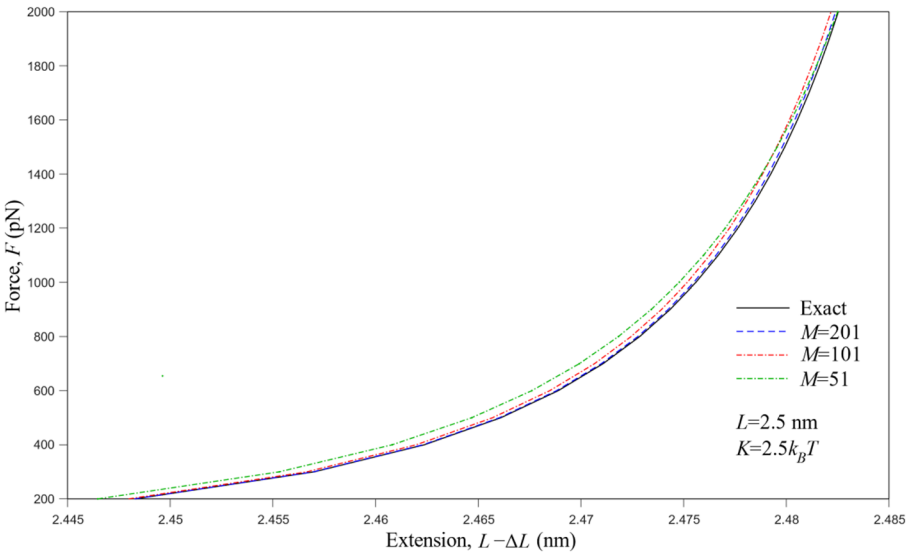


Fig. 31 Force-extension relations in a rod under thermal fluctuation at $T = 300°\mathrm{K}$

8 Conclusions

The study presents a unified approach for interpolation and regression of data with irregularities and scatter in a multi-dimensional space. It provides a single mathematical framework for diverse datasets and multi-dimensional data manipulation and model order reduction. The mathematical framework is based on the Peridynamic Differential Operator (PDDO) to transfer information within a set of discrete data, and among data sets representing multiple scales. The robustness and capability of this approach have been demonstrated by considering various real or fabricated data concerning two- or three-dimensional applications. The numerical results concern interpolation, regression, and recovery/compression of non-uniform data and model reduction.

Appendix 1 Peridynamic differential operator

According to the 2-order TSE in a 2-dimensional space, the following expression holds

$$f(\mathbf{x}+\xi) = f(\mathbf{x}) + \xi_1 \frac{\partial f(\mathbf{x})}{\partial x_1} + \xi_2 \frac{\partial f(\mathbf{x})}{\partial x_2} + \frac{1}{2!}\xi_1^2 \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} + \frac{1}{2!}\xi_2^2 \frac{\partial^2 f(\mathbf{x})}{\partial x_2^2} + \xi_1\xi_2 \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} + R \tag{97}$$

where R is the small remainder term. Multiplying each term with PD functions, $g_2^{p_1 p_2}(\xi)$, and integrating over the domain of interaction (family), $H_{\mathbf{x}}$, results in

$$\begin{aligned}\int_{H_x} f(\mathbf{x}+\xi) g_2^{p_1p_2}(\xi) dV = f(\mathbf{x}) \int_{H_\mathbf{x}} g_2^{p_1p_2}(\xi) dH_{\mathbf{x}'} + \frac{\partial f(\mathbf{x})}{\partial x_1} \int_{H_\mathbf{x}} \xi_1 g_2^{p_1p_2}(\xi) dH_{\mathbf{x}'} \\ + \frac{\partial f(\mathbf{x})}{\partial x_2} \int_{H_\mathbf{x}} \xi_2 g_2^{p_1p_2}(\xi) dH_{\mathbf{x}'} + \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} \int_{H_\mathbf{x}} \frac{1}{2!} \xi_1^2 g_2^{p_1p_2}(\xi) dH_{\mathbf{x}'} \\ + \frac{\partial^2 f(\mathbf{x})}{\partial x_2^2} \int_{H_\mathbf{x}} \frac{1}{2!} \xi_2^2 g_2^{p_1p_2}(\xi) dH_{\mathbf{x}'} + \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} \int_{H_\mathbf{x}} \xi_1 \xi_2 g_2^{p_1p_2}(\xi) dH_{\mathbf{x}'}\end{aligned} \tag{98}$$

in which the point $\mathbf{x}$ is not necessarily symmetrically located in the domain of interaction. The initial relative position, ξ, between the material points $\mathbf{x}$ and $\mathbf{x}'$ can be expressed as $\xi = \mathbf{x} - \mathbf{x}'$. This ability permits each point to have its own unique family with an arbitrary position. Therefore, the size and shape of each family can be different, and they significantly influence the degree of non-locality. The degree of the interaction between the material points in each family is specified by a non-dimensional weight function, $w(|\xi|)$, which can vary from point to point. The interactions become more local with a decreasing family size. Thus, the family size and shape are important parameters. In general, the family of a point can be non-symmetric due to non-uniform spatial discretization. Each point has its own family members in the domain of interaction (family), and occupies an infinitesimally small entity such as volume, area, or a distance.

The PD functions are constructed such that they are orthogonal to each term in the TS expansion as

$$\frac{1}{n_1! n_2!} \int_{H_\mathbf{x}} \xi_1^{n_1} \xi_2^{n_2} g_2^{p_1p_2}(\xi) dV = \delta_{n_1p_1} \delta_{n_2p_2} \tag{99}$$

with $(n_1, n_2, p, q = 0, 1, 2)$ and $\delta_{n_ip_i}$ is a Kronecker symbol.

Enforcing the orthogonality conditions in the TSE leads to the non-local PD representation of the function itself and its derivatives as

$$f(\mathbf{x}) = \int_{H_\mathbf{x}} f(\mathbf{x}+\xi) g_2^{00}(\xi) dH_{\mathbf{x}'} \tag{100}$$

$$\left\{ \begin{array}{c} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \frac{\partial f(\mathbf{x})}{\partial x_2} \end{array} \right\} = \int_{H_\mathbf{x}} f(\mathbf{x}+\xi) \left\{ \begin{array}{c} g_2^{10}(\xi) \\ g_2^{01}(\xi) \end{array} \right\} dH_{\mathbf{x}'} \tag{101}$$

$$\left\{ \begin{array}{c} \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_2^2} \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} \end{array} \right\} = \int_{H_\mathbf{x}} f(\mathbf{x}+\xi) \left\{ \begin{array}{c} g_2^{20}(\xi) \\ g_2^{02}(\xi) \\ g_2^{11}(\xi) \end{array} \right\} dH_{\mathbf{x}'} \tag{102}$$

The PD functions can be constructed as a linear combination of polynomial basis functions:

$$\begin{aligned} g_2^{p_1p_2}(\xi) = a_{00}^{p_1p_2} w_{00}(|\xi|) + a_{10}^{p_1p_2} w_{10}(|\xi|) \xi_1 + a_{01}^{p_1p_2} w_{01}(|\xi|) \xi_2 \\ + a_{20}^{p_1p_2} w_{20}(|\xi|) \xi_1^2 + a_{02}^{p_1p_2} w_{02}(|\xi|) \xi_2^2 + a_{11}^{p_1p_2} w_{11}(|\xi|) \xi_1 \xi_2 \end{aligned} \tag{103}$$

where $a_{q_1q_2}^{p_1p_2}$ are the unknown coefficients, $w_{q_1q_2}(|\xi|)$ are the influence functions, and ξ_1 and ξ_2 are the components of the vector ξ. Assuming $w_{p_1p_2}(|\xi|) = w(|\xi|)$ and submitting the PD functions into the orthogonality equation lead to a system of algebraic equations for the determination of the coefficients as

$$\mathbf{Aa} = \mathbf{b} \tag{104}$$

in which

$$\mathbf{A} = \int_{H_{\mathbf{x}}} w(|\xi|) \begin{bmatrix} 1 & \xi_1 & \xi_2 & \xi_1^2 & \xi_2^2 & \xi_1\xi_2 \\ \xi_1 & \xi_1^2 & \xi_1\xi_2 & \xi_1^3 & \xi_1\xi_2^2 & \xi_1^2\xi_2 \\ \xi_2 & \xi_1\xi_2 & \xi_2^2 & \xi_1^2\xi_2 & \xi_2^3 & \xi_1\xi_2^2 \\ \xi_1^2 & \xi_1^3 & \xi_1^2\xi_2 & \xi_1^4 & \xi_1^2\xi_2^2 & \xi_1^3\xi_2 \\ \xi_2^2 & \xi_1\xi_2^2 & \xi_2^3 & \xi_1^2\xi_2^2 & \xi_2^4 & \xi_1\xi_2^3 \\ \xi_1\xi_2 & \xi_1^2\xi_2 & \xi_1\xi_2^2 & \xi_1^3\xi_2 & \xi_1\xi_2^3 & \xi_1^2\xi_2^2 \end{bmatrix} dH_{\mathbf{x}'} \tag{105}$$

$$\mathbf{a} = \begin{bmatrix} a_{00}^{00} & a_{00}^{10} & a_{00}^{01} & a_{00}^{20} & a_{00}^{02} & a_{00}^{11} \\ a_{10}^{00} & a_{10}^{10} & a_{10}^{01} & a_{10}^{20} & a_{10}^{02} & a_{10}^{11} \\ a_{01}^{00} & a_{01}^{10} & a_{01}^{01} & a_{01}^{20} & a_{01}^{02} & a_{01}^{11} \\ a_{20}^{00} & a_{20}^{10} & a_{20}^{01} & a_{20}^{20} & a_{20}^{02} & a_{20}^{11} \\ a_{02}^{00} & a_{02}^{10} & a_{02}^{01} & a_{02}^{20} & a_{02}^{02} & a_{02}^{11} \\ a_{11}^{00} & a_{11}^{10} & a_{11}^{01} & a_{11}^{20} & a_{11}^{02} & a_{11}^{11} \end{bmatrix} \tag{106}$$

and

$$\mathbf{b} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{107}$$

After determining the coefficients $a_{q_1q_2}^{p_1p_2}$ via $\mathbf{a} = \mathbf{A}^{-1}\mathbf{b}$, then the PD functions $g_2^{p_1p_2}(\xi)$ can be constructed. The detailed derivations and the associated computer programs can be found in [8].

Appendix 2 Ordinary Kriging

As described by Cressie [4], the observed (known) data at spatial locations, $s_1, ..., s_n$, are modeled as a random process denoted by $Z(s)$ in ordinary Kriging. It is assumed that the random process satisfies the decomposition

$$Z(s) \equiv \mu + \delta(s) \tag{108}$$

where μ is the expected value or mean of $Z(s)$ and $\delta(s)$ is the correlated error process. The mean, μ, is unknown; however, it is assumed to be a constant. The predictor is defined as

$$p(Z;B) = \sum_{i=1}^{n} \lambda_i Z(s_i) \tag{109}$$

subject to the constraint of

$$\sum_{i=1}^{n} \lambda_i = 1 \tag{110}$$

The parameter, B, is defined as a block of spatial area whose location and geometry are known. The optimal predictor is obtained by minimizing the mean-squared prediction error defined as

$$\sigma_e^2 \equiv E(Z(B) - p(Z;B))^2 \tag{111}$$

with respect to the coefficients λ_i with $i = 1, ..., n$.

Appendix 3 Temperature Data from Weather Stations in Arizona

Weather station name	Elevation (m)	Latitude	Longitude	Temp. (°F)
Robson Ranch	455.7	32.8118	−111.6313	65
Tacna 3 NE	98.8	32.7225	−113.9191	68
Fry	2194.6	35.07	−111.84	52
Gunsight	1609.3	36.7044	−112.5833	50
Williams	2105.9	35.2413	−112.1929	56
Guthrie	1097.3	32.8819	−109.3092	53
Rincon	2511.6	32.2056	−110.5481	60
Frazier Wells	2063.5	35.8456	−113.055	52
Bisbee 1 WNW	1694.7	31.4475	−109.9288	61
Tempe ASU	355.7	33.4258	−111.9216	71
Wittmann 1 SE	513	33.7776	−112.523	71
Painted Desert National Park	1755.6	35.068	−109.7688	50
Union Pass	1072.9	35.2247	−114.3747	62
Humbug Creek	1600.2	34.1164	−112.3006	60
Hurricane	1659.6	36.6992	−113.2072	47
Saguaro National Park	938.8	32.1794	−110.7363	57
Willcox	1271	32.2553	−109.8369	58
Oak Creek	1500.8	34.9417	−111.7517	56
Greer	2499.4	34.06	−109.45	57
Safford Agricultural Center	900.4	32.815	−109.6808	63
Betatakin	2220.8	36.6778	−110.5411	44
Tohono Chul	770.2	32.3391	−110.9808	63
Empire	1417.3	31.7806	−110.6347	69
Four Springs	1999.5	36.7939	−112.0422	41
Teec Nos Pos	1612.4	36.9233	−109.09	42
Patagonia Paton Center	1232.6	31.53923	−110.76028	72

Weather station name	Elevation (m)	Latitude	Longitude	Temp. (°F)
Payson	1516.4	34.2431	−111.3028	61
Chiricahua	1645.9	32	−109.35	62
Paria Point	2205.2	36.7278	−111.8219	49
Cherry	1554.5	34.5964	−112.0481	58
Sanders	1784	35.2239	−109.3222	52
San Carlos Reservoir	771.8	33.1819	−110.5261	46
Goodwin Mesa	1280.2	34.75	−113.3	63
Sunset Crater National Monument	2127.5	35.3694	−111.5436	42
Sunrise Mountain	2856	33.9733	−109.563	42
Flagstaff	2133.6	35.145	−111.675	56
Hopi	1885.5	35.8103	−110.2069	53
Selles	721.2	31.91	−111.8975	69
Iron Springs	1804.4	34.5853	−112.5019	61
Hopkins	2170.2	31.6753	−110.88	66
Tucson International Airport	776.9	32.1313	−110.9552	70
Mormon Mountain	2286	34.94	−111.52	53
Catalina State Park	825.1	32.4177	−110.9302	61
Sunset Point	902.2	34.1953	−112.1417	64
Tweeds Point	1585	36.5819	−113.7319	51
Douglas Bisbee Inter. Airport	1251.2	31.4583	−109.6061	66
St. Johns Industrial Air Park	1747.4	34.51833	−109.37917	52
Nixon Flats	1981.2	36.39	−113.1522	54
Black Rock	2158	36.7944	−113.7567	45
Tusayan	2042.2	35.99	−112.12	55
Stray Horse	2139.7	33.5406	−109.3169	56
Snowslide Canyon	2965.7	35.34	−111.65	47
Stanton	1097.3	34.1667	−112.7333	64
Olaf Knolls	883.9	36.5072	−113.8161	60
Kingman Airport	1042.4	35.2577	−113.933	63
Douglas	1231.4	31.345	−109.5394	66
Nogales 6 N	1054.9	31.4554	−110.968	73
Truxton Canyon	1630.7	35.7825	−113.7942	56
Lindbergh Hill	2682.2	36.2858	−112.0794	48
Bagdad	1199.4	34.5975	−113.1745	57
White Horse Lake	2188.5	35.14	−112.15	56
Alamo Dam	393.2	34.228	−113.5777	65
Walnut Creek	1551.4	34.9281	−112.8097	62
Walnut Canyon National Monument	2040.6	35.1721	−111.5097	43
Dry Lake	2264.1	33.3597	−109.8331	60
Casa Grande	426.7	32.8875	−111.7147	67
Duncan	1115.6	32.748	−109.1213	57
Dry Park	2653.6	36.45	−112.24	49
Heber Black Mesa Ranger Station	2008.6	34.3925	−110.558	56
Alpine	2447.8	33.8417	−109.1222	57
Prescott Love Field	1536.8	34.65167	−112.42083	62
Phoenix Airport	337.4	33.4277	−112.0038	68

Weather station name	Elevation (m)	Latitude	Longitude	Temp. (°F)
Benson 6 SE	1124.7	31.8805	−110.2403	58
Yuma Proving Ground	98.8	32.8356	−114.3942	68
Winslow Airport	1489.3	35.0281	−110.7208	51
Fort Valley	2240.3	35.27	−111.74	56
Happy Jack Ranger Station	2279.9	34.7433	−111.4139	47
Bellemont Weather Forecast Office	2179.9	35.2302	−111.8221	50
Beaver Dam	588.6	36.9139	−113.9423	60
Sasabe	1094.2	31.483	−111.5436	75
Show Low Airport	1954.1	34.2639	−110.0075	50
Window Rock Airport	2054	35.6575	−109.06139	52
Moss Basin	1804.4	35.0336	−113.8925	58
Picacho 8 SE	603.8	32.6463	−111.4017	63
Jerome	1508.8	34.7522	−112.1114	49
Quartzsite	266.7	33.665	−114.2272	69
Page Municipal Airport	1313.7	36.92611	−111.44778	43
Warm Springs Canyon	2441.4	36.7	−112.23	51
Kitt Peak	2069.6	31.96018	−111.59787	60
Workman Creek	2103.1	33.81	−110.92	55
Yellow John Mountain	1877.6	36.1542	−113.5417	54
Carefree	771.1	33.8161	−111.9019	67
Grand Canyon National Park Airport	2013.5	35.94611	−112.15472	49
Montezuma Castle Nat.Monument	969.3	34.6105	−111.838	59
Scottsdale Municipal Airport	449	33.62278	−111.91056	68
Havasu	144.8	34.7872	−114.5617	69
Limestone Canyon	2072.6	34.1789	−110.2736	51
Kartchner Caverns	1429.5	31.8352	−110.3552	53
Buckskin Mountain	1950.7	36.9306	−112.1997	48
Mount Lemmon Willow Canyon	2141.8	32.3859	−110.69799	47
Apache Junction 5 NE	630.9	33.4625	−111.4813	66
Anvil Ranch	840.9	31.9793	−111.3837	63
Canyon de Chelly	1709.9	36.1533	−109.5394	47
Green Valley	883.9	31.893	−110.9977	59
Bright Angel Ranger Station	2438.4	36.2147	−112.0619	41
Organ Pipe Cactus Nat. Monument	511.5	31.9555	−112.8002	73
Fort Huachuca Pioneer Airfield	1453.3	31.60722	−110.42806	66
Globe	1112.5	33.3503	−110.6519	56
San Simon	1091.2	32.29329	−109.22691	61
Phantom Ranch	771.1	36.1066	−112.0947	58
Petrified Forest National Park	1659.9	34.7994	−109.885	49
Saint Johns	1764.8	34.5172	−109.4028	49
Meteor Crater	1687.1	35.0364	−111.0231	47
Music Mountain	1652	35.6147	−113.7939	56
Muleshoe Ranch	1272.5	32.4	−110.2708	67
Tombstone	1420.1	31.7119	−110.0686	56
Sierra Vista	1403.9	31.53699	−110.28073	53
Hilltop	1743.5	33.6183	−110.42	61

Weather station name	Elevation (m)	Latitude	Longitude	Temp. (°F)
Elgin 5 S	1466.4	31.5907	−110.5087	67
Ajo	533.7	32.3698	−112.8599	66
Smith Peak	762	34.1158	−113.3472	65

Funding This study was performed as part of the ongoing research at the MURI Center for Material Failure Prediction through Peridynamics at the University of Arizona (AFOSR Grant No. FA9550-14-1-0073).

Declarations

Conflict of Interest The authors declare no competing interests.

References

1. Franke R (1982) Scattered data interpolation: Tests of some methods. Math Comput 38:181–200
2. Mittas L, Mitasova H (1999) Spatial interpolation. In: Longley PI, Goodchild MF, Maguire DJ, Rhind DW (eds) Geographical Information Systems: Principles. Wiley, Techniques, Management and Applications, GeoInformation International, pp 481–492
3. Steffensen JF (2006) Interpolation, 2nd edn. Dover Publications
4. Cressie N (2015) Statistics for spatial data, Revised. John Wiley & Sons
5. Liszka T, Orkisz J (1980) The finite difference method at arbitrary irregular grids and its applications in applied mechanics. Comp Struct 11:83–95
6. Liszka T (1984) An interpolation method for irregular net of nodes. Int J Numer Meth In Engr 20:1599–1612
7. Madenci E, Barut A, Futch M (2016) Peridynamic differential operator and its applications. Comp Meth Appl Mech Eng 306:408–451
8. Madenci E, Barut A, Dorduncu M (2019) Peridynamic differential operator for numerical analysis. Springer, New York
9. Madenci E, Barut A, Dorduncu M (2019) Peridynamics for predicting thermal expansion coefficient of grapheme. 69th Electronic Components & Technology Conference, Las Vegas, NV
10. Madenci E, Barut A, Prohit P (2020) A Peridynamic approach to computation of elastic and entropic interactions of inclusions on a lipid membrane. J Mech Phys Solids 104046
11. https://www.ncdc.noaa.gov/cdo-web/search?datasetid=GHCND
12. https://the.me/the-story-of-lena/
13. Su T, Purohit PK (2010) Thermomechanics of a heterogeneous fluctuating chain. J Mech Phys Solids 58:164–186
14. Boal D (2002) Mechanics of the cell. Cambridge University Press, Cambridge, UK
15. Zhang Y, Crothers DM (2003) Statistical mechanics of sequence-dependent circular DNA and its application for DNA cyclization. Biophys J 84:136–153